

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-131145

(43)Date of publication of application : 13.05.1994

(51)Int.Cl. G06F 3/14

G06F 12/00

G06F 15/00

(21)Application number : 04-131137 (71)Applicant : CANON INC

(22)Date of filing : 22.05.1992 (72)Inventor : WATANABE TETSUSHI

YOSHIKAWA NAOHIRO

HIKITA YOSHIRO

TAKADA YASUFUMI

(54) DATA PROCESSOR

(57)Abstract:

PURPOSE: To provide a data processor by which a user can execute processing even if he has no knowledge related to data intended to process and the detail of a program to process it.

CONSTITUTION: The data being controlled is stored in hierarchial structure in an HDD 9. Besides, a network interface 11 is connected to another computer system. Information such as the attribute, etc., of the data, the attribute of each unit of data management, and the correspondence between the data and the program capable of processing it are stored in the HDD 9 together with the data. In the case that the data and the program are displayed by an icon, they are displayed on a CRT 1 so that a

processable pair can be discriminated on the basis of the information. Besides, at the time of the retrieval of the data too, the user is never burdened in order to obtain the stored place, etc., of the data in the information.

**\* NOTICES \***

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

---

**CLAIMS**

---

[Claim(s)]

[Claim 1]A data processing device which is used by two or more users in a built-in application program, and manages data per user under hierarchical managing structure, comprising:

A kind of processing which can be performed with said this application program.

A management tool which manages matching with data managed in said user unit which is the target of this processing.

A means to express processing by said application program, and said data by a respectively peculiar icon.

A displaying means which displays correspondence with processing with said application program and data of a processing object of this program based on said management tool by an icon peculiar to each.

[Claim 2]The data processing device according to claim 1, wherein said displaying means carries out highlighting of an icon of processing with said corresponding application program, and the icon of data of a processing object of this program.

[Claim 3]The data processing device according to claim 1, wherein said displaying means displays a line which connects this icon with an icon of data of processing with said corresponding application program, and a processing object of this program.

[Claim 4]The data processing device according to claim 1 piling up said displaying means and redisplaying it when an icon of processing with said corresponding application program and an icon of data of this processing object approach within a

predetermined distance.

[Claim 5]The data processing device according to claim 1 beyond prescribed distance's detaching said displaying means, and redisplaying it when an icon of processing with said corresponding application program and an icon of data other than this processing object approach within a predetermined distance.

[Claim 6]A data processing device which manages data per user under hierarchical managing structure characterized by comprising the following.

A management tool which manages the attribute of data.

A search condition setting-out means to set up a search condition for search.

A search means to search data corresponding to conditions which are under an attention hierarchy's data and were set up by said search condition from data covering a desired hierarchy number.

A search-results presenting means which shows a user a result of said search means  
\*\*\*\* search.

[Claim 7]The data processing device according to claim 6 being able to refer to data corresponding to a search condition without having further a copying means which copies DETA \*\* corresponding to a search condition obtained by said search means to an attention hierarchy with the attribute of this data, and following a hierarchical structure.

[Claim 8]The data processing device according to claim 7 performing data processing for data which was further provided with a means to start a program which performs data processing, and was searched by said search means.

[Claim 9]A data processing device which manages data per user under hierarchical managing structure characterized by comprising the following.

A structure display means to display said data-hierarchy managing structure.

A search condition setting-out means to set up conditions for searching data.

A search means to search data corresponding to conditions which are under an attention hierarchy's data and were set up by said search condition setting-out means from data covering a desired hierarchy number.

A displaying means which displays specially said low order data corresponding to said search condition when displaying a search-results presenting means which shows a user search results by this search means, and low order data under data obtained by said search according to the managing structure.

[Claim 10]A data processing device which manages data for every user under hierarchical managing structure characterized by comprising the following.

A control means which operates said data.

A management tool which manages information of a user allowed operation of said data.

A search condition setting-out means to set up conditions which search data.

A search means to search data corresponding to conditions which are under an attention hierarchy's data and were set up by said search condition from data covering a desired hierarchy number, and a retrieval execution control means which controls execution of search by said search means based on information of a user by said management tool.

[Claim 11]A user management tool which manages a user allowed operation of data based on said control means as a user's set, The data processing device according to claim 10 having further the second retrieval execution control means that controls execution of retrieval processing by said search means based on information on a set of a user by said user management tool.

[Claim 12]A data processing device which manages data for every user under hierarchical managing structure characterized by comprising the following.

A control means which operates said data.

A management tool which manages information of a user allowed operation of said data.

A search condition setting-out means to set up conditions which search data.

A search means to search data corresponding to conditions which are under an attention hierarchy's data and were set up by said search condition from data covering a desired hierarchy number, and a displaying means which displays a result of retrieval processing by said search means based on information of a user by said management tool.

[Claim 13]A user management tool which manages a user allowed operation of data based on said control means as a user's set, The data processing device according to claim 12 having further the second displaying means that displays a result of retrieval processing by said search means based on information on a set of a user by said user management tool.

---

## DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001]

[Industrial Application]This invention relates to the data processing device etc. with which a user can operate desired data, for example on two or more data processing devices with two or more users.

[0002]

[Description of the Prior Art]Conventionally, in the data processing device with two or more users, the user's data is saved in given places, such as a memory defined by the data processing device. When one user was going to operate the data corresponding to specific conditions, a procedure peculiar to the data processing device has realized operation of data. For example, when an application program tries to have performed data manipulation, a procedure peculiar to the application program has realized. When the data which is the target of operation existed on other data processing devices connected in a certain form, it has realized by taking the procedure as which specifying the data processing device with which the target data exists etc. was determined.

[0003]With the data processing device used by two or more conventional users, each user's data was stored in the predetermined memory location, and when it was going to operate the data corresponding to desired conditions, the memory location, specification of conditions, etc. have been realized in a procedure peculiar to a data processing device.

[0004]

[Problem(s) to be Solved by the Invention]However, in the above-mentioned conventional example, peculiar professional expertise was needed for every data processing device, and the user had to master the complicated operation for every computer system. When the data which exists on other data processing devices was operated, it had to be known on which processing unit the data would exist. For example, when application performs data manipulation, a user, It had to be known beforehand where [ the kind of data which know hows, such as processing which can be performed for every application program, are needed, and can be referred to or updated, or ] it is stored. The user had to know beforehand conversely the application program which can refer to or update it for every data, and the kind of processing.

[0005]This invention was accomplished in view of the above-mentioned conventional example, and it aims at providing the data processing device which can operate desired data, without a user being conscious of it, wherever data may be stored in.

[0006]It aims at providing the data processing device with which a user does not need to be conscious of correspondence with data and the application program which can process it.

[0007]

[Means for Solving the Problem]It reaches. [Function]To achieve the above objects, the data processing device of this invention consists of the following composition.

[0008]Are a data processing device which is used by two or more users in the built-in application program, and manages data per user under hierarchical managing structure, and The kind of processing which can be performed with said this application program, The management tool which manages matching with the data managed in said user unit which is the target of this processing, A means to express processing by said

application program, and said data by a respectively peculiar icon, It has a displaying means which displays correspondence with the processing with said application program and the data of the processing object of this program based on said management tool by an icon peculiar to each.

[0009]The management tool which is a data processing device which manages data per user under hierarchical managing structure, and manages the attribute of data, A search means to search the data corresponding to the conditions which are under an attention hierarchy's data and were set to a search condition setting-out means to set up the search condition for search, by said search condition from the data covering a desired hierarchy number, It has a search-results presenting means which shows a user the result of said search means \*\*\*\*\* search.

[0010]A structure display means to be a data processing device which manages data per user under hierarchical managing structure, and to display said data-hierarchy managing structure, A search means to search the data corresponding to the conditions which are under an attention hierarchy's data and were set to a search condition setting-out means to set up the conditions for searching data, by said search condition setting-out means from the data covering a desired hierarchy number, When displaying the search-results presenting means which shows a user the search results by this search means, and the low order data under the data obtained by said search according to the managing structure, it has a displaying means which displays specially said low order data corresponding to said search condition.

[0011]The control means which is a data processing device which manages data for every user under hierarchical managing structure, and operates said data, The management tool which manages the information of the user allowed operation of said data, and a search condition setting-out means to set up the conditions which search data, It is under an attention hierarchy's data and has a search means to search the data corresponding to the conditions set up by said search condition, and a retrieval execution control means which controls execution of search by said search means based on the information of the user by said management tool from the data covering a desired hierarchy number.

[0012]The control means which is a data processing device which manages data for every user under hierarchical managing structure, and operates said data, The management tool which manages the information of the user allowed operation of said data, and a search condition setting-out means to set up the conditions which search data, It is under an attention hierarchy's data and has a search means to search the data corresponding to the conditions set up by said search condition, and a displaying means which displays the result of the retrieval processing by said search means based on the information of the user by said management tool from the data covering a desired hierarchy number.

[0013]

[Example]

<Explanation of device> drawing 1 is a block diagram of the data processing device in an example. In the figure, 1 shows CRT which displays information, a various-messages menu, etc. of data under processing with an application program. 2 shows the Video RAM (the following, VRAM) which develops the image displayed on the screen of CRT1. The bit processing unit in which 3 controls the data transfer between memories or between a memory and each device. (The following and BMU) are shown, 4 is the keyboard provided with the various keys for processing a text editing etc., and 5 shows the pointing device (following, PD) used in order to point to the icon on a picture, etc. 6 is CPU which manages control of this whole device, and 7 is ROM which has memorized the program etc. which realize the operation processing procedure of CPU6. The program concerning the flow chart which begins the application program and error-handling program concerning data processing, and shows them by drawing 26 etc. which are mentioned later is also memorized by this ROM7. 8 shows RAM used as a work area, momentary evacuation area at the time of error handling, etc., when CPU5 executes the various programs mentioned above. 9 shows a hard disk drive (following, HDD), 10 shows a floppy disk drive (following, FDD), respectively, and it is used for the preservation later mentioned on each disk, such as an application program, data, and a library. 11 shows a network interface (following, Net-I/F), and in order to perform data transfer via a network among other data processing devices, it performs data control on a network, and diagnosis. 12 is an I/O bus (it consists of an address bus, a data bus, and a control bus) which connects between each unit mentioned above.

[0014]The data processed in <conceptual of desktop management system> this example by the application programs which process data, and those application programs is managed, The program which processes giving the environment where the data is operated to a user, and giving data to execution of an application program and the executed application program etc. is called a "desktop management system." As a display example of a actual desktop management system, as drawing 3 shows, it is built in two or more windows on a windowing system, and data is operated by the input from the keyboard 4, the pointing device 5, etc.

[0015]Next, various concepts in a desktop management system are explained.

[0016]Drawing 2 expresses about the room 13 and the office 14 which are the concepts of a user's work environment in a desktop management system. In the room 13, are a concept to point out the space which provides the environment which one user owns, and in the office 14. It is a concept which points out the space which comprises a set of the room 13 contained in it, and the room which is the work environment of the worker individual who works [ an office and there ] in the actual world is imitated.

[0017]For example, two or more rooms 13 exist, respectively every office 14 which

distributed the physical file system of one computer as the one office 14 to two or more computers combined in the network. About the function to provide the work environment included in the room 13, reference is made by explanation of the below-mentioned operation objects. One user becomes possible [ performing starting of an application program and processing of data ] by having one's room 13 in the suitable office 14, receiving service of a desktop management system within the room 13, and operating the below-mentioned operation objects. The user can access the room 13 of other users of the same office 14 besides his room 13, or the room 13 of other offices 14, and can do the same operation as the inside of his room 13 within allowed limits. It is also possible to operate movement of data, a copy, etc. ranging over two or more rooms 13.

[0018]Operation objects (object) show the data expressed as an icon on a window and the data aggregate, and the function which a desktop management system provides further, in order to provide the accessing means to the desktop management system by a user.

[0019]Next, the operation objects provided by a user interface using drawing 2 and each figure are explained. In drawing 2, although various objects belong to the room "Takada", a room "Sato" or the room "Watanabe" is the same.

[0020]Drawing 3 shows the room window G01 showing a user's room 13 realized on a windowing system. The object which is in the room window G01 using drawing 2 and drawing 3 is explained.

[0021]The desktop 15 shown by the desktop object G02 is an object which shows the function to manage the application program which is one function of a desktop management system. The window containing the below-mentioned object in a desktop is displayed by open operation of the desktop G02.

[0022]The cabinet 16 shown by the cabinet object G03 is an object which points out the entry of the top level of a function which manages the data created by the application program by a layered structure. The window containing the below-mentioned object in a cabinet is displayed by open operation of the cabinet object G03.

[0023]The library 17 shown by the library object G04 is an object which points out the entry of the place where the library whose application program is ancillary data (it is common) referred to when processing data is stored. The window containing the below-mentioned object in a library is displayed by open operation of the library object G04.

[0024]The tag tray 19 shown by the tag tray object G05 is an object showing the virtual entry for accessing the binder 20 and the data 18 which are mentioned later, without following the layered structure of a cabinet. The tag 21 points out the object to store. The window containing the below-mentioned object in a tag tray is displayed by open operation of the tag tray object G05.



[0025]In the open shelf 22 shown by the open shelf object G06. Operation for permitting that other users refer to or update the 16 or less-cabinet data 18 and the binder 20 in a user's own room 13 (open operation is called hereafter) is realized, and the object which saves the information is pointed out. The window containing the below-mentioned object in an open shelf is displayed by open operation of the open shelf object G06.

[0026]The garbage can 23 shown by the garbage can object G07 points out the object which manages abandonment of data. The clipboard 24 shown by the clipboard object G08 is an object which provides the function for exchanging the data 18 which the application 26 processes, or its part between the applications 26.

[0027]The exit 25 shown by the exit object G09 is an object showing the accessing means to other rooms 13 and offices 14 of a user. The window containing the below-mentioned object in an exit is displayed by open operation of the exit object G09.

[0028]Although it is except operation objects, The title as which G10 displays a user's name and the title of a window currently displayed, A manual operation button for G11 to direct the operation in a window, the help button in which G12 displays the help contents of operation, and G13 express the scroll bar which is made to scroll the icon outside a window and is displayed.

[0029]Drawing 4 shows the desktop window G14 in the state where it was opened by the desktop G02. The operation objects in the desktop window G14 are explained using drawing 4.

[0030]The application 26 shown by the application object G15 points out the object for operating an application program (starting). Operation of the application object G15 is mentioned later.

[0031]Drawing 5 shows the cabinet window G16 in the state where it was opened by the cabinet object G03. The operation objects in the cabinet window G16 are explained using drawing 5.

[0032]The binder 20 shown by the binder object G17 is an object which shows the node under the cabinet 16 which manages data by a layered structure.

It can create arbitrarily under the cabinet 16 and the binder 20.

The window containing the binder 20 and the data 18 in the binder 20 is displayed by open operation of the binder object G17.

[0033]The data 18 shown with the data object G18 is an object showing the data 18 created by the application 26, and it is used in order to operate the data 18 or to start the application program which processes the data 18. Operation of the data object G18 is mentioned later. The object in the binder 20 is also the same.

[0034]Drawing 6 shows the library window G19 in the state where it was opened by the library object G04. The operation objects in the library window G19 are explained using drawing 6.

[0035]The library binder 27 shown by the library binder object G20 is an object for storing collectively the library data 28 of the same kind managed as a library mentioned later. G21 is a numerical value as which the library data 28 under the library binder 27 expresses the number of times referred to by the application program now. The window which contains the data 18 in the opened library binder 27 by open operation of the library binder object G20 is displayed.

[0036]Drawing 7 shows the library binder window G22 in the state where it was opened by the library binder object G20. The operation objects in the library binder window G22 are explained using drawing 7.

[0037]In the library data 28 shown with the library data object G23, it is an object which points out what is especially managed as the above-mentioned library among the data 18 created by application. The library data 28 is stored in the library binder 27 with same thing of the same data format.

[0038]Drawing 8 shows the tag tray window G24 in the state where it was opened by the tag tray object G05. The operation objects in the tag tray window G24 are explained using drawing 8.

[0039]The tag 21 shown by the tag object G25 is an object displayed virtually, in order to access the object in a cabinet of the binder 20 and the data 18, without following the layered structure of a cabinet. The same operation as binder [ of the substance to which the tag 21 is pointing to the tag 21 ] 20, and data 18 object is possible.

[0040]Drawing 9 shows the open shelf window G26 in the state where it was opened by the open shelf object G06. The operation objects in the open shelf window G26 are explained using drawing 9.

[0041]The open tray 29 shown by the open tray object G27 points out the object created by every [ of other users who become an open object, or a user ] set (an user group is called hereafter), when a user performs open operation. The window containing the open object 30 is displayed by open operation of the open tray object G27.

[0042]Drawing 10 shows the open tray window G28 in the state where it was opened by the open tray object G27. The operation objects in the open tray window G28 are explained using drawing 10. The open object G29 displays virtually the object of the cabinet 16 of the user to whom open operation was given, the binder 20, and the data 18. The user or user group which serves as an object which exhibits the open object G29 with the attribute of the open tray 29 in which the open object G29 exists is determined. Open operation is mentioned later.

[0043]Drawing 61 shows the clipboard window G62 in the state where it was opened by the clipboard object G08. In the clipboard window G62, the clip data object G63 which is operation objects which show the clip data 32 is contained. They are the data 18 created by the application 26 or its part, and data containing the format ID in the clip data 32. The clip data 32 generated by the application 26 is stored in the clipboard

24 in the user's room 13, and is displayed clip data object G63 in the clipboard window G62. Operation of the clipboard 24 is mentioned later.

[0044]Drawing 11 shows the exit window G30 in the state where it was opened by the exit object G09. The operation objects in the exit window G30 are explained using drawing 11. The door object G31 is an object showing the entrance to the room of other users who exist in the office where a user belongs. The user name is displayed on the door object G31 which shows the door 31, The window of a user's room 13 which the door 31 shows by carrying out open operation of the door object G31 is displayed, The object which shows the cabinet 16, the binder 20, and the data 18 in which the user of the room 13 performed open operation to the user group including a user or a user is displayed. By operating the exhibited object, the user can access the user's data.

[0045]The office object G32 is an object showing the entrance to other offices. The office name is displayed on the office object G32, and door 31 object showing the entry to the room contained by the open operation in the office is displayed. The operation to this door 31 is the same as that of said door 31. The library object G33 is a library which the office where a user belongs manages. Operation and the contents are the same as the contents explained by above-mentioned drawing 6. The clipboard object G34 is a clipboard which the office where a user belongs manages. Operation and the contents are the same as the contents explained by above-mentioned drawing 61.

[0046]The management information needed when the <information for desktop management systems> desktop management system operates is explained.

[0047]Drawing 14 is a figure showing the table of the data information 50. The data information 50 is information for registering and managing the form of the data 18 which the application program 26 which the desktop 15 manages creates, and a kind, and the following item is included about each data format. 51 is data format ID and is a peculiar number for each data format of every. 52 is an extension, and when creating the file of the data 18 of the data format, it is a character attached to the last of a file name. 53 is an icon-files name and expresses the file name in which the data of the icon used when displaying the data 18 in the binder 20 is stored. 54 expresses library ID which is ID of that library data and which is mentioned later, when the data of this data format can be managed also as library data. It is possible to register two or more library ID.

[0048]Drawing 15 is a figure showing the table of the application information 60. The application information 60 is information about the application 26 which the desktop 15 manages.

Based on this information, the object of the application 26 is displayed in the window of the below-mentioned desktop 15.

The following information is included every registered application 26. 61 is application

ID and is the peculiar number were received and numbered for each application program of every. 62 is an execution module name and expresses the name of the program used when performing application 26. 63 is an application name and is displayed as a name of the object of the application 26. 64 is an icon-files name and expresses the file name in which the data of the icon used when carrying out the icon display of the application 26 is stored. 65 is a processing list in which correspondence with format ID51 which shows the data format of the data 18 which can process the application program, a process code, and the name for displaying the processing is shown. 66 is an entry of the head of the processing list 65 in every format ID51 among the processing lists 65, This expresses default processing of the application 26, shows processing when the application 26 is started without specifying the data 18 used as a processing object, and consists of format ID51 and its processing list. It expresses whether when 67 performs processing under processing list, the data 18 used as a processing object is required for it, and if it is "O", data is necessity "x" and it is unnecessary and blank, it is shown that whichever may be sufficient.

[0049]Drawing 16 is a figure showing the table of the library information 70. The library information 70 is information about the library data 28 which a desktop management system manages.

Based on this information, the library binder 27 is displayed in the window of the below-mentioned library 17.

The following information is included every registered library binder 27. 71 is library ID and is the peculiar number were received and numbered every library binder 27. 72 is a library binder name and shows the whereabouts (directory) of the library binder 27 in the file system of the layered structure where the library data 28 is saved. 73 is a name of the library which is the data aggregate with the same data. 74 expresses now the number of times opened to the library data 28 under the library binder 27.

[0050]Drawing 17 is a figure showing the table of User Information 80. User Information 80 is information which manages the user of the system by which one desktop management system works.

The following information is included for every user.

81 is user ID and is the peculiar number were numbered to each user. 82 expresses the arbitrary information about users, such as a user's name. 83 is the whereabouts (path of a directory) of a user's room 13. 84 expresses the updating mode of the library data 28 in a user's room 13. This library update mode is mentioned later.

[0051]Drawing 18 is a figure showing the table of the office information 90. The office information 90 is information which manages the office 14 where other desktop management systems connected in the network etc. are working.

The following information is included every office 14.

91 is office ID and is the peculiar number each office was numbered to 14. 92 is a host name in which the office 14 exists. 93 is an office name and expresses the name of

the office 14 which exists in a host. 94 is a domain and expresses a host's domain where the office 14 belongs. 95 expresses the updating mode of the library data 28 in an office. A domain means a set of the computer with the same architecture, between the desktop management systems which work on the computer belonging to the same domain, does not have to carry out special conversion and can transmit the data which the application program created.

[0052]Drawing 19 is a figure showing the table of the group information 100. Since the range of the user for restricting reference of the data 18 and updating is expressed, the group information 100 is information which manages the group who is the unit which summarized two or more users.

The following information is included for every group.

101 is group ID and is the peculiar number were received and numbered for every group. 102 is the user ID of the user belonging to each group.

[0053]<Cabinet administrative information> drawing 20 is a figure showing the table of the object management information 110. The object management information 110 is a file for managing the information on each object contained in the binder 20, and exists every binder 20 in the cabinet 16 (the library 17 and the garbage can 23 are also included) and the bottom of it. The object management information 110 exists in order [ per office 14 / one ] to manage the library data 28 of the office 14. As for the data management information 110, the following item is included about one data. 111 shows the kind (the binder 20, the data 18, the open shelf 22, the open tray 29, the tag tray 19, the tag 21, the library binder 27, the library data 28) of object. 112 expresses the file name or directory name which is the contents of the object. 113 shows format ID of the data, when an object is data or library data. When an object is a library binder in the library 17, the library ID is shown. When an object is data, 114 is application ID of the application 26 which created the data, and calls especially this creator ID. 115 is a Chinese character name of an object. 116 is the size of the byte unit of the data, when an object is data, and when data comprises two or more files (fragmentation: after-mentioned), it expresses the size of the sum total. 117 is a registered date and the object expresses the date created newly. 118 is with an updated date and the object expresses the date changed at the end. 119 is with a reference day and the object expresses the date referred to at the end.

[0054]120 expresses the user ID of the owner of an object. 121 expresses the group ID of the owner of an object. 122 expresses the user ID of the user who referred to the object at the end. 123 expresses the user ID of the user who updated the object at the end. 124 is an access level, classifies the object of the permission of reference and update operation and the disapproval to an object into three kinds, the owner user ID 120 of data, the user of the group 121 to whom the owner of data belongs, and the other user, and is set up.

[0055]125 expresses the domain of the office 14 where the data was created /

updated, when an object is data. 126 expresses the user ID of the object user of the open tray, i.e., the user who becomes an open object, when an object is the open tray 29 exhibited to other users. 127 expresses the group ID of the candidate group of the open tray, i.e., the group who becomes an open object, when an object is the open tray 29 exhibited to other groups. 128 expresses the public presentation which is a setting detail which restricts the copying operation of the object to the user of a public presentation place, when an object is the open object 30. 129 expresses the open term which is a term when open operation of the open object 30 is effective, when an object is the open object 30. 130 expresses the data abandonment term which is a term automatically discarded with the garbage can 23, when an object is the placed data in the garbage can 23, and data carries out fixed organization progress. The tag reference destination 131 shows the whereabouts of the operation objects (substance) to which the tag 21 is pointing, when an object is the tag 21.

[0056]Drawing 21 is a figure showing the table of the open object management information 140. The open object management information 140 is the binder 20 opened now in each user's room 13, and information which manages the state of the data 18.

One exists every room 13.

[0057]The following item is included in the open object management information 140 about one object. 141 is a file name of the object. 142 is the user ID of the user who opens the object. It is a flag showing whether 143 is opened by the object by "reference" of the access level 124, or it is opened by "updating." 144 expresses the date opened by the object.

[0058]Drawing 22 is a figure showing the library search order information 150. The library search order information 150 is it information showing the turn of the library 17 at the time of searching the library data 28, and one exists every room 13.

[0059]<Opening of binder> drawing 23 is a figure showing the operation which opens the binder 20 in the cabinet 16. The cursor G37 indicates the position on the screen of the pointing device 5 to be among drawing 23, G38 is a mark showing button grabbing of the pointing device 5 for an operation description, and it means that the button was pushed by button 1 click "\*" by "\*" at button 2 click and "\*", and that the button was detached by "\*\*." These signs are used through an operation description. If open operation of the binder 20 moves the cursor G37 to the icon of the binder object G17 in the cabinet window G16 and is clicked button 2, the binder window G39 which displays the contents of the binder 20 will appear. This open operation is operation common to operation objects at large.

[0060]If the flow chart of drawing 24 explains the open shop operation of the binder 20, The object management information 110 on the binder 20 which the binder object G17 clicked two times by S1 shows is acquired, Icon files are read from format ID113 of

the object management information 110 acquired by S2, and the data information 50, and an icon display is performed on the binder window G39 generated by S3. Deed processing is ended for these processings to all the entries of the object management information 110 by S4. This processing is common to the cabinet 16, the open shelf 22, the tag tray 19, and the library binder 27. The open shop operation of the desktop 15, the exit 25, and the library 17 is described later.

[0061]<Copy of data> drawing 25 is a figure showing the copy operation of the data 18 between the cabinet 16 or the binder 20. The copy operation of the data 18 moves the cursor G37 to the icon of the data object G18 of the copying origin to wish, It completes by operation (drag / drops operation) of moving to the cabinet window G16 or the binder window G39 of a copy destination which wishes the cursor G37 with a button pushed, and detaching a button.

[0062]If drawing 26 explains the copy processing method of the data 18 in a cabinet, the object management information 110 on the data 18 will be acquired from the binder 20 with which the data 18 of a copied material exists in the stage where the button was pushed on the icon of a copied material by S10. When the data 18 of the acquired object management information 110 to a copied material is the open object 30 and there is setting out of the open copy restrictions 128 by S11, deed processing is stopped by S12 for the display of the purport of copy prohibition. After a button is detached by S13 in a copy destination, it inspects whether the copy of files, such as disk storage capacity, is possible at S14, and if a copy is impossible, deed processing will be stopped by S15 for an error display. The entry of the object management information 110 beforehand acquired by S16 by the object management information 110 on the binder 20 of a copy destination is added, and processing is ended by copying the file which is the contents of the data 18 in S17.

Application is started when a user operates the object which the application 26 managed by a <starting of application> desktop management system or the data 18 shows. Starting of application, Execution module name [ of application ] 62, file name [ of the data 18 processed following it ] 112, and format ID51, It performs by adding the character string which directs the disposal method under processing list 65 (example: /usr/local/bin/ezps/ezps-f 012-me).

[0063]The application object G15 which shows the application 26 which is opened by the desktop window G14 and has already been registered like drawing 4 by drawing 3 if the desktop object G02 in the room window G01 is clicked two times is displayed. The open shop operation of the desktop 15 is ended by acquiring the application information 60, reading icon files, and performing an icon display on a window.

[0064]With application, opening of the data 18 will add the entry of the data 18 opened in the open object management information 140 of the room in which the data 18 exists. If the data 18 is closed by application, the entry of the data 18 opened by the above-mentioned open object management information 140 will be deleted.

[0065]Next, the starting processing of the application in a desktop management system is explained according to a user's operation using figures.

[0066](1) Starting by open operation of an application object.

[0067]By drawing 27, the application object G15 in the desktop window G14 is clicked two times.

[0068]If the flow chart of drawing 28 explains, the application information 60 of the application 26 which the application object clicked two times by S20 shows will be acquired, Format ID and the process code of the default processing 66 of the head entry of the processing list 65 are passed by S21, and an application program is started.

[0069](2) Starting by open operation of an application object.

[0070]If the application object G15 in the desktop window G14 is chosen by the button 1 click of a mouse by drawing 29, appearance, like the frame of an icon becomes thick will change so that having been chosen can distinguish the application object G15. If "operation" button G40 is chosen by button 1 click from the menu lists G41 in this state, "operation" menu G42 will be displayed. Processing to perform with the application 26 shown by the selected application object G15 is chosen from displayed "operation" menu G42.

[0071]If the flow chart of drawing 30 explains, when "operation" button G40 is not chosen in S30, other processings will be performed by S31. When "operation" button G40 is chosen, The application information 60 of the application 26 chosen by S33 after the application object G15 was chosen by S32 is acquired, It means that format ID in the head entry of the processing list 65 of the application information 60 acquired by S34 was chosen as a default, and the processing name of the processing list 65 of above-mentioned format ID is displayed on "operation" menu G42. About the function which needs the already created data of a print function etc., by the processed-data necessity 67 for the application information 60, it is inspected by "operation" menu G42 and is not displayed on it. If one item of "operation" menu G42 is chosen by S35, the process code corresponding to format ID which is S36 and was already chosen as a default, and the selected item will be passed, The application program which the already selected application object G15 shows is started.

[0072](3) Starting by the drag / drops operation of an application object.

[0073]By drawing 31, drag of the application object G15 in the desktop window G14 is carried out, and it drops in the cabinet (binder) window G16. The application information 60 of the application 26 which drag / dropped application object G15 shows as well as the case of (1) is acquired, Format ID in the head of the processing list 65 of the application information 60 and the process code of the default processing 66 are passed, and an application program is started.

[0074](4) Starting by open operation of a data object.

[0075]By drawing 32, the data object G18 in the cabinet (binder) window G16 is



clicked two times. If the flow chart of drawing 33 explains, creator ID114 and format ID113 of the data will be obtained from the object management information 110 on the data 18 shown with the data object G18 clicked two times by S40.

[0076]Obtain the entry of the application information 60 which has the application ID61 [ same ] as the above-mentioned profitable \*\*\*\* creator ID114 by S41, and by S42 Next, the above-mentioned profitable \*\*\*\* format ID113, The process code of the default processing 66 of above-mentioned profitable \*\*\*\* format ID in the entry of the above-mentioned profitable \*\*\*\* application information 60 and the file name 112 of data are passed, and an application program is started.

(5) Starting by open operation of a data object.

[0077]If the data object G18 in the cabinet (binder) window G16 is chosen by button 1 click by drawing 34, the appearance of an icon being highlighted so that having been chosen can distinguish the data object G18 will change. If "operation" button G40 is chosen by button 1 click from the menu lists G41 in this state, "operation" menu G42 will be displayed. Processing to perform to the data 18 shown in the data object G18 chosen from displayed "operation" menu G42 is chosen.

[0078]If the flow chart of drawing 35 explains, when "operation" button G40 on a window is not chosen, other processings will be performed by S51. When "operation" button G40 is chosen, the data object G18 is chosen by S52, Format ID113 and creator ID114 of the data 18 which the selected data object G18 shows are obtained from the object management information 110, The entry which waits for the application ID61 [ same ] as creator ID114 is obtained from the application information 60, The processing name of the processing list 65 which corresponds to format ID113 obtained from the entry of the above-mentioned profitable \*\*\*\* application information 60 by S54 is displayed as "operation" menu G42. When one item of "operation" menu G42 is chosen by S55, by S56 The file name 112 of the data 18, The process code corresponding to format ID113 already obtained and the item chosen from "operation" menu G42 is passed, and an application program is started by the execution module name 62 of the above-mentioned profitable \*\*\*\* application information 60.

[0079](6) Carry out the drag/drops of the data object on an application object, and start it.

[0080]By drawing 36, drag of the one data object G18 in the cabinet (binder) window G16 is carried out, and it drops with the application object G15 in the desktop window G14.

[0081]If the flow chart of drawing 37 explains and drag of the data object G18 will be carried out, file name [ of the data 18 ] 112 and format ID113 shown with the selected data object G18 will be obtained from the object management information 110 by S60. When the data object G18 by which drag was carried out by S61 is located on the application object G15 of the desktop window G14, by S62. Acquire the application

information 60 of the located application 26, and by S63. If format ID of the data 18 in which the application 26 is shown with the data object G18 by which drag is carried out can be processed, The method of presentation of reversing the icon display of the above-mentioned application object G15 by S64 is changed, and it tells that the data 18 which the data object G18 in which drag of the application 26 is carried out to the user shows can be processed. If the data object G18 by which drag was carried out by S65 is dropped, It is inspected whether it was dropped on the application object G15 of the application 26 which can process the above-mentioned data 18 by S66, The process code of the default processing 66 of format ID applicable to application ID61 is obtained from the above-mentioned profitable \*\*\*\* application information 60 by S67, Format ID113, a process code, and the file name 112 of the data 18 are passed to the application 26 shown by the execution module name 62, and an application program is started to it.

[0082]In this desktop management system, as drawing 15 explained, When the data 18 of one data format can be processed with two or more applications 26, and when the one application 26 can process the data 18 of two or more data formats, the relation between the data 18 and the application which can process it is managed. For this reason, it is useful to tell the user who is operating it about the ability of selection or the data object G18 which carried out drag to be dropped [ with which application object G15 ], i.e., which application 26 can process that data 18?. The following guidance function is provided as a means to realize this.

[0083](6.1) Guidance of combination by a line.

[0084]By drawing 38, when the glove (unite for pointing cursor) of the one data object G18 in the cabinet window G16 or the binder window G39 is carried out, The data object G18 and the application object G15 in which the processing in the desktop window G14 is possible are connected by a line, and it shows like drawing 39.

[0085]If the flow chart of drawing 40 explains this processing and the glove of the data object G18 will be carried out, file name 112 and format ID113 will be obtained from the data management information 110 by S70 about the data 18 corresponding to that data object G18. The application 26 in the desktop 15 is searched in S71. In S72, if it is judged that the application 26 does not exist besides having already inspected, presentation of the application which can be processed will be ended. When application exists, in S73, the application information 60 about the application 26 is acquired. In S74, it is judged whether there is format ID of the data 18 which the data object G18 by which the glove is carried out into the application information 60 shows. A certain data object by which the glove was carried out when becoming can be processed with the application found out by S71.

[0086]If the application object G43 of the desktop window G14 can process the data object G18, As shown in drawing 39, the data object G18 and the application object G43 are connected with S75 by the line G44, The application 26 which the application

object G43 shows to a user tells that the data 18 which the data object G18 by which the glove is carried out shows can be processed. If the application 26 which returns control of processing to S71 and has not been inspected yet by S72 exists after presentation processing of S75 is completed about the one searched application 26 in S71, Processing of S73 to S75 is henceforth repeated about the application 26 in the desktop 15. The data object G18 in which the glove was carried out in the cabinet window G16 by this as drawing 39 showed, for example after processing was completed by judgment of S72, The application objects G43 and G45 of the application 26 which can process it are connected with the lines G44 and G46.

[0087](6.2) Guidance of combination by highlighting.

[0088]By drawing 41, when the glove of the one data object G18 in the cabinet window G16 or the binder window G39 is carried out, The display of highlighting the application object G15 in the desktop window G14 which can process the data object G18 is changed and shown. This processing is explained to be drawing 41 with the flow chart of drawing 42.

[0089]If the glove of the data object G18 is carried out, the file name 112 and format ID113 will be obtained from the data management information 110 by S80 about the data 18 corresponding to the data object G18. The application 26 in the desktop 15 is searched in S81. In S82, if it is judged that the application 26 does not exist besides having already searched, presentation of the application which can be processed will be ended. When application exists, in S83, the application information 60 about the application 26 is acquired. In S84, it is judged whether there is format ID of the data 18 which the data object G18 by which the glove is carried out into the application information 60 shows. A certain data object by which the glove was carried out when becoming can be processed with the application found out by S81. For example, if the application object G47 of the desktop window G14 can process the data object G18, By S85, as shown in drawing 41, highlighting of the display of the application object G47 is carried out, The application 26 which the application object G47 shows to a user tells that the data 18 which the data object G18 by which drag is carried out shows can be processed. If the application 26 which returns control of processing to S81 and has not been inspected yet by S82 exists after presentation processing of S85 is completed about the one application 26 searched in S81, Processing of S83 to S85 is henceforth repeated about the application 26 in the desktop 15. By this, after processing is completed by judgment of S82, For example, highlighting of the application objects G47 and G48 of the application 26 which can process the data object G18 by which the glove was carried out in the cabinet window G16 as drawing 41 showed is carried out.

[0090](6.3) Guidance by attraction.

[0091]When carrying out drag of the one data object G18 in the cabinet window G16 or the binder window G39 and moving it by drawing 43, When the data object G18

which is carrying out drag reaches the distance defined from the application object G15 in which the processing in the desktop window G14 is possible, The display position of the data object G18 by which drag was carried out is amended automatically, and a desktop management system shows it. The flow chart of drawing 43 and drawing 44 explains this processing.

[0092]If the glove of the data object G18 is carried out, the file name 112 and format ID113 will be obtained from the data management information 110 by S90 about the data 18 corresponding to the data object G18. In S91, it is judged whether the overlapping area of the data object G49 and the application object G50 by which drag was carried out reached the fixed rate with the display surface product of the data object G49 by which drag was carried out. In S91, when it is judged that the area of a lap with the icon of the data object G49 by which drag was carried out, and the application object G50 has not reached a fixed rate with the display surface product of the data object G49 by which drag was carried out, control is returned to S91.

[0093]In S91, the area of a lap with the icon of the data object G49 by which drag was carried out, and the application object G50, If it is judged that the fixed rate with the display surface product of the data object G49 by which drag was carried out was reached, the application information 60 of the application 26 which the application object G50 shows in S92 will be acquired.

[0094]In S94, it is tested whether format ID of the data 18 which the data object G49 by which drag is carried out shows is in the application information 60 of the application 26 which the application object G50 shows. If it is, and data can be processed with the overlapping applications and it will be judged that it is possible, In S95, the display position of the data object G49 by which drag was carried out is amended, like G51, it is made for the data object G49 and the application object G50 to lap thoroughly, and they carry out highlighting of the application object G50 by S96 simultaneously. A user senses that the data object G49 which is carrying out drag was able to draw near to the application object G50 with attraction by this processing. It can judge [ that the data 18 which the data object G49 in which the application 26 which the application object G50 shows is carrying out drag shows can be processed, and ].

[0095]In S97, it is judged following S96 whether the data object G49 by which drag was carried out was dropped on the application object G50. In S97, when it is judged that it is not dropped, control is returned to S91. When it is judged in S97 that it was dropped, In S98, from the application information 60 acquired by S92. The process code of the default processing 66 of format ID applicable to application ID61 is obtained, The file name 112 of the data 18 which format ID, a process code, and the data object G18 show to the application 26 which an application object shows is passed, and the application 26 is started.

[0096](6.4) Guidance by reaction force.

[0097]When carrying out drag of the one data object G18 in the cabinet window G16 or the binder window G39 and moving it by drawing 45, When the data object G18 which is carrying out drag reaches the distance defined from the application object G15 in which the processing in the desktop window G14 is impossible, The display position of the data object G18 which carried out drag is amended automatically, and a desktop management system shows it. This processing is explained to be drawing 45 with the flow chart of drawing 88.

[0098]If the glove of the data object G18 is carried out, file name 112 and format ID51 will be obtained from the data management information 110 by S90 about the data 18 corresponding to the data object G18. In S91, it is judged whether the area of the lap of the data object G53 and the application object G52 by which drag was carried out reached the fixed rate with the display surface product of the data object G53 by which drag was carried out.

[0099]In S91, when it is judged that the area of a lap with the icon of the data object G53 by which drag was carried out, and the application object G52 has not reached a fixed rate with the display surface product of the data object G52 by which drag was carried out, control is returned to S91. In S91, the area of a lap with the icon of the data object G53 by which drag was carried out, and the application object G52, If it is judged that the fixed rate with the display surface product of the data object G52 by which drag was carried out was reached, the application information 60 of the application 26 which the application object G52 shows in S92 will be acquired.

[0100]In S94, to the application information 60 of the application 26 which the application object G52 shows. Control will be put into starting of the same application 26 as the following (6.3), if there is format ID of the data 18 which the data object G53 by which drag is carried out shows and it is judged [ that the data 18 can be processed and ]. If the application 26 which the application object G52 shows is judged [ that the data 18 which the data object G53 by which drag is carried out shows cannot be processed and ] in S94 on the contrary, In S93, the display position of the data object G53 by which drag was carried out is amended, and the lap of the data object G53 and the application object G52 is decreased or abolished like G54. A user senses that the data object G53 which is carrying out drag was pulled apart from the application object G52 by reaction force by this processing, It can judge [ that the data 18 which the data object G53 in which the application 26 which the application object G52 shows is carrying out drag shows cannot be processed, and ].

[0101](7) Drag/drop two or more data objects on an application object, and start them.

[0102]By drawing 38, drag of two or more data objects G18 in the cabinet (binder) window G16 is carried out, and it drops with the application object G15 in the desktop window G14. Format ID51 of each data object G18 by which drag was carried out is obtained like (6). If the data object G18 by which drag was carried out is located on the application object G15 of the desktop window G14, Noting that it can process, if

there is format ID of all the data objects G18 which search the application information 60 and are dragged to the application 26 which the application object G15 shows, It tells that the data 18 which all the data objects in which the method of presentation of highlighting the icon display of the application object G15 is changed, and drag of the application 26 is carried out to the user show can be processed.

[0103]If two or more data objects G18 by which drag was carried out are dropped on the application object G15 by which highlighting was carried out before, The application information 60 is searched from format ID51 of the data 18 which application ID61 of the application 26 which the application object G15 shows, and a data object show, The process code of the default processing 66 of execution module name 62 and format ID51 applicable to application ID61 is obtained, the file name 112 of format ID51, a process code, and two or more data 18 is passed, and the application 26 is started.

[0104]<Open operation> Other users are not usually allowed to update with reference to the cabinet 16 in a user's room 13, the binder 20, and the data 18. Operation of permitting a user referring to or updating the cabinet 16, the binder 20, and the data 18 which he owns to other users is called "open operation." The cabinet 16, the binder 20, and the data 18 which were performed or it performed "open operation" are called the open object 30. Next, the operation and processing for performing "open operation" are explained.

[0105]First, the open tray 29 is created on the open shelf 22 for every user who considers it as an open object, or group. A preparation method chooses New button G55 on a window on the open shelf window G26 in drawing 46, An input of the name or the group ID 101 of partner's user's name opened to the input window G56 of an inquiry, the user ID 81, or a group will create and display the open tray object G27 which shows the new open tray 29 in the open shelf window G26.

[0106]Then, the open tray object G27 of the user for public presentation or a group is opened, and the open tray window G28 is displayed. Drag of the data object G18 or the binder object G17 to exhibit is carried out like the copy operation of the data 18 between the binders 20 here, and it drops with up to the open tray window G28. It means that the open object G29 dropped by this operation was displayed on the open tray window G28, and open operation was performed. It is possible to set up setting out which restricts other users' copying operation to the exhibited object, and the term which permits public presentation.

[0107]Setting operation is clicking the open object G29 to set up after pushing the manual operation button G57 of the open tray window G28 in drawing 47 one time, and since the input window G58 of an inquiry is displayed, setting out completes it in inputting a setting-out item. When setting out has already been carried out, a setting detail is displayed in a setting-out item field at the time of input window G58 display. The flow chart and drawing 46 of drawing 48 explain open tray 29 creation processing

by a desktop management system.

[0108]New button G55 on the open shelf window G26 creates the input window G56 which inputs an open object by selection \*\*\*\*\* and others [ S100 ], When it is inputted by waiting S101 and the input from a user is inputted with an user name in S102, User Information 80 is searched with S103, and the user ID 81 is obtained. When inputted by the user ID 81 in S104, it is inspected whether the user ID 81 which searched User Information 80 with S105, and was inputted exists. When inputted with a group name in S106, the group information 100 is searched with S107, and the group ID 101 is obtained. When inputted by the group ID 101 in S108, it is inspected whether the group ID 101 which searched the group information 100 with S109, and was inputted exists.

[0109]When the user ID 81 or the group ID 101 exists by S110, the entry of the new open tray 29 is created to the object management information 110 on the open shelf 22 by S111, The user ID 81 or the group ID 101 before obtained by the public presentation place user ID 126 or the public presentation place group ID 127 is substituted. The open tray object G27 which wets the open tray 29 created in S112 is displayed in the open shelf window G26, and processing is ended. Processing is stopped when the user ID 81 or the group ID 101 is not obtained.

[0110]Next, the flow chart and drawing 47 of drawing 49 explain "open operation" processing by a desktop management system.

[0111]When drag of the object to exhibit is carried out, the object management information 110 on the object by which drag was carried out in S120 is acquired. When the object by which drag was carried out by S121 is dropped on the open tray window G28, The object management information 110 which creates the entry of the new open object 30 to the object management information 110 on the open tray 29 which the open tray window G28 shows, and is acquired in front by S122 is substituted. By S123, it displays on the open tray window G28 by making into the open object G29 the object by which drag was carried out, and processing is ended.

[0112]Next, the flow chart of drawing 47 and drawing 50 explains open copy restriction setting out and the open term setting processing to the open object 30 in a desktop management system.

[0113]First, if the manual operation button G57 on the open tray window G28 is chosen by S130, the object management information 110 on the open object G29 chosen in the selection input of the open object G29 waiting and S133 will be acquired by S132.

[0114]Next, the input window G28 of an inquiry is created by S134, the contents of the open copy restrictions 128 and the open term 129 are displayed from the object management information 110 before acquired on the input column, and it waits for an input by S135. It substitutes for the object management information 110 acquired with the contents near at hand of the open copy restrictions 128 of having been inputted

by S136, and the open term 129 after a setting input, the object management information 110 which setting out completed is substituted to the object management information 110 in the open tray 29, and processing is ended.

[0115]Drawing 2 explains operation and processing in case a <access to other users' data> user accesses other users' data.

[0116]First, opening of the exit 25 in its room 13 will display the library 17 which the door 31 showing the room 13 of other users in their office 14, other offices 14, and their office 14 manage. If other offices 14 are opened, the office's 14 the door 31 and the library 17 of a user which were opened will be displayed again. Opening of the door 31 of the user who wishes will display the open object 30 to which the user of the opened room 13 has performed open operation to himself. The same operation as the operation objects in its room 13 is possible for the operation to this open object 30. However, when setting out of open copy restrictions is carried out to the open object 30, performing operation which copies the contents of the open object 30 is forbidden. When setting out of the open term is carried out to the open object 30, the open object 30 which passed over the open term is not displayed on a window.

[0117]The flow chart of drawing 51 explains the exit 25 open shop operation in a desktop management system to drawing 3, drawing 11, drawing 12, and the drawing 13 pan.

[0118]If opened by the exit object G09, the exit window G30 will be displayed and other users and the name of the office 14 will be obtained from User Information 80 and the office information 90 by S140. The library object G33 which shows the door object G31 which shows the door 31, the office object G32 which shows the office 14, and the library 17 is displayed on the exit window G30 by S141, and it waits for an input by S142.

[0119]When open operation is performed to the library object G33 by S143, deed processing is ended for the open shop operation of the library 17 later mentioned by S144. When open operation is performed to the office object G32 by S145, By requiring User Information 80 of the host by whom the host name 92 was opened by obtaining from the office information 90 by S146, acquire User Information 80 and it returns to S141, The door object 30 of the door 31 which shows the entry to a user's room contained in the office is displayed on the office window G35 of drawing 12, and it waits for an input.

[0120]When open operation is performed to the door object G31 by S147, the object management information 110 on the user public presentation shelf 22 of the door 31 opened by S148 is acquired, and the open tray 29 for users currently operated by S149 is searched. When the open tray 29 for users currently operated exists, only the open object 30 which acquires the object management information 110 on said open tray 29 by S150 and over which the open term has not passed by S151 is displayed on the room window G44 of drawing 13. When the open tray 29 does not exist by S149,



that is displayed by S125 and processing is ended.

[0121]When the <tag> tag 21 operates a hierarchy's deep binder 20 or data 18, it is an object which makes it possible to operate the target binder 20 or the data 18, without performing open operation of the intermediate hierarchy binder 20 two or more times.

[0122]Next, the generation method and disposal method of the tag 21 are explained.

[0123]Opening of the tag tray object G05 which shows the tag tray 19 of drawing 3 will display the tag tray window G24 where the tag object G25 was displayed like drawing 8. The actual condition of the displayed tag object G25 is the binder 20 and the data 18 which exist under a "cabinet", and the processing as the operation to the binder 20 and the data 18 which are the actual condition that the operation to the tag 21 which the displayed tag object G25 shows is the same is performed. However, tag 21 self is deleted about deleting operation, and the binder 20 or the data 18 which is the actual condition is not deleted. The new production method of the tag 21, It completes with carrying out drag of the binder object G17 of the binder 20 or the data object G18 of the data 18 which is the actual condition of the tag 21 to create like the data 18 copy operation between the binders 20, and dropping with the tag tray window G24. It means that the tag object G25 which displayed the icon information of the binder object G17 drag was carried out on the tag tray window G24 by this operation of, or the data object G18 was displayed, and new production of the tag 21 was carried out. The flow chart of drawing 52 explains new production processing of the tag 21.

[0124]When drag of the binder object G17 or the data object G18 is carried out by S160, the object management information 110 on the binder 20 that the data 18 which the binder 20 which the binder object G17 shows, or the data object G18 shows exists is acquired. When a button is detached on the tag tray window G24 by S161, the entry of the object management information 110 before acquired by the object management information 110 on the tag tray 19 which the tag tray window G24 shows by S162 is added. Furthermore the name of the binder 20 with which the binder 20 or the data 18 exists in the tag reference binder name 131 within the object management information 110 is substituted for S163, the tag object G25 created by S164 is displayed on the tag tray window G24, and processing is ended.

[0125]Next, the flow chart of drawing 53 explains open operation processing of the tag object G25.

[0126]When operation is performed to the tag object G25, the object management information 110 on the tag 21 which the operated tag object G25 which exists in the tag tray 19 by S170 shows is acquired. The object management information 86 on the binder name 131 in which the binder 20 or the data 18 which is the actual condition in S171 exists and the binder 20, or the data 18 is acquired.

[0127]When said object management information is acquired by S172, the processing as the treating operation to the binder 20 or the data 18 that subsequent processing is the same at S173 is performed. When said object management information is not

acquired by S172, deed processing is ended for the display of the purport that the data or the binder which a tag shows by S174 does not exist.

In order to discard the binder 20 and the data 18 of which <abandonment of data> creation was done, Like the copy operation of the data 18 between the binders 20, By open operation of the garbage can object G07 displayed on the room window G01 of drawing 3. The binder object G17 and the data object G18 move to a garbage can window with dropping the binder object G17 of the binder 20 to discard in the garbage can window G59 displayed, and the data object G18 of the data 18. In the desktop management system, the process of supervising data abandonment is always operating, and the moved binder object G17, the binder 20 which it shows, and the data object G18 and the data 18 which it shows are eliminated after fixed time preservation.

[0128]The flow chart of drawing 5, drawing 54, and drawing 55 explains the discarding treatment of data.

[0129]When the glove of the data object G18 of the data 18 is carried out by S180, the object management information 110 on the binder 20 that the data object G18 exists is acquired. When a button separates on the garbage can window G59 by S181 and the object management information 110 before acquired by S182 has setting out of updating disapproval, deed processing is ended for a display of that it cannot discard in S183.

[0130]If it is cases other than \*\*\*\*, if it is not opened by others by the data which it is going to discard or is tested and opened by S184, abandonment will be improper and it will progress to S183, and if not opened, copy processing will be performed for the above-mentioned data 18 to the garbage can 23 in S185. When copy processing is normally completed by S186, the entry of said data 18 of the object management information 110 on the binder 20 that said data 18 exists by S187 is deleted, and the file of said data 18 is also eliminated.

[0131]Finally the date after fixed time is substituted for S188 from the present time in the data abandonment term of the entry of said data 18 added to the object management information 110 on the garbage can 23, and processing is ended.

[0132]The flow chart of drawing 56 explains the disposal method of a data abandonment supervisory process. First, the object management information 110 on the garbage can 23 is acquired by S190. The entry of the object management information 110 over which the data abandonment term has passed to each entry is searched with S191. By deleting the file which the file name in the above-mentioned entry and the above-mentioned entry expresses with S192 when it exists, The data 18 or the binder 20 which the deleted above-mentioned entry shows is deleted, and, as for the data object G18 or the binder object G17, the garbage can window G59 is not displayed.

[0133]Whether these processings were performed for all the entries of the

above-mentioned object management information 110 by S193, and after testing and all ending, it returns carrying out fixed time progress to waiting and S190 in S194 and S195, and the same processing is repeated.

[0134]The data managed to a <library management> library refers to the ancillary data for referring to it in common with other applications, in order that an application program may process data. For example, when the application program which processes document data is made into an example, to document data. The font data for displaying format data with the information for the command for operating a document orthopedically being included, interpreting the command as library data, and operating a document orthopedically with alphabetic data and its alphabetic data, or its alphabetic data is managed as independent data. An application program processes one document data by this, When displaying on CRT or carrying out a printout, the appearance can be changed easily, without adding change to document data by changing the font data as library data and format data which are used, for example as ancillary data. Also when using one format data in common with two or more document data, since it becomes unnecessary to include format data into each document data, hypertrophy of document data can be prevented. The data in the state where it is managed to these libraries is caught with the concept of the library data 28.

[0135]the information which specifies the required library data 28 is added to the data 18, and required of the application 26 and its data 18 to be used -- it can lend library data 28, and is decided and required. Also when the application 26 creates the data 18 newly, the existing library data 28 can be used.

[0136]Next, management of a library is explained according to figures.

[0137]The library data 28 of the form of data 18 and object management information 110 the very thing managed under the cabinet 16 which the library data object G23 displayed in the library binder window G22 shows like drawing 7 is the same. These are also manageable under the cabinet 16 as the usual data 18. When a desktop management system manages the data 18 managed under the cabinet 16 also as the library data 28, The management as the library data 28 holds only the (the user attached) name, and when the library data 28 is needed, the name may refer the substance and it may be made to refer to the data 18 under the cabinet 16 by a name.

[0138]Like drawing 3 and drawing 11, the library 17 which a desktop management system manages has a library displayed in the library displayed in the user's room window G01, or the exit window G30. The library expressed by the library object G33 displayed on the exit window G30 is a library attached to the office 14 to which the room 13 belongs.

The library expressed by the library object G04 displayed in a user's room window G01 is a library attached to the room 13.

Thus, the library 17 is classified into the following two according to the gestalt of the

management.

[0139](1) The library 17 attached to the room 13.

[0140]In the personal library 17 registered as an object of the room 13, the useful range serves as one user. The useful range of the library 17 can also be considered as two or more users' set specified arbitrarily by enabling it to access from other users by performing the same processing as open operation of the above-mentioned data 18.

[0141](2) The library 17 attached to the office 14.

[0142]In the library 17 registered as an object of the office 14, only the user of the room 13 contained in the office 17 is the library 17 which can be accessed in common, and the useful range serves as two or more users' set everlastingly specified by the system. The useful range of the library 17 can also be made into all the users by the above-mentioned open operation.

[0143]The library 17 has managed the library binder 27 to every [ which is shown in drawing 16 ] library information 70 (kind of library data 28).

Opening of the library 17 will display the library window G19 like drawing 6.

Only the library data 28 with format ID51 [ same ] is stored in the one library binder 27. The library binder 27 of library ID71 for which it waits as an attribute is the same as that of library ID54 of the data information 50 of the library data 30 stored.

[0144]A <creation of library binder> user explains to the library 17 in the room 13 the procedure and processing which create the new library binder 27 using the flow chart of drawing 57 and drawing 58.

[0145]When a user creates the new library binder 27 in the library 17 in the room 13, the generation button G60 of the library binder of the library window G19 is chosen by S200. A desktop management system obtains only an entry with registration of library ID54 from data information by S202, and displays the library selected window G61 which displayed the list of only the above-mentioned data information 50 in the library selected window 101 in S203. Library ID71 by which new production was carried out to the data information 50 chosen [ above-mentioned ] by S205 by a user choosing one of them S204 by carrying out additional creation of the entry of the library information 70 in the library 17 is substituted, An additional indication of the library binder object G20 of the library binder 27 is given on the window of the library 17 by S206.

[0146]When registering the application 26 newly and a new library is required, a required group's (library data 31 classification) library binder 27 and library data 28 can be registered. At this time, the library information 70 is registered into a desktop management system with the group of that library 17 in registration of the application 26, The default data 18 which the application 26 provides at this time is registered as the library 17 attached to the office 14. If the group of the library 17 is registered, the library binder 27 is generated for every group of the, and, i.e., registered as the library data 28, and the group's data will be managed.

[0147]Operation and the procedure of registering the data 18 under the <registration to library of data> cabinet 16 as the library data 28 are explained using the flow chart of drawing 59.

[0148]When registering the data 18 under the cabinet 16 as the library data 28, it becomes the same operation as the copy (movement) of the usual data 18 shown in drawing 25. A user opens the library binder object G20 which shows the library binder 27 of the library 17 of a registration destination, and displays the library binder window G22. Drag of the data object G18 showing the data 18 in the binder 20 registered into a library is carried out, and it drops in the library binder window G22.

[0149]A desktop management system acquires the object management information 110 on the binder 20 of a copied material by S210, is S211 next and obtains the entry of the data information 50 with the format ID51 [ same ] as format ID113 of the object management information 110 acquired before. Furthermore by S212, library ID71 of the library binder 27 which the library binder window G22 where the data object G18 was dropped shows is obtained from the library information 70 of the library 17. Library ID54 and library ID71 of the data information 50 which were obtained by S211 are compared, when it differs, the operation which cannot be registered by S214 is displayed, and processing is stopped by S213.

[0150]When library ID54 and library ID71 are in agreement by S214, Copy processing which already explained the data 18 to the library binder 27 from the binder 20 of the copied material by S215 is performed, and the data 18 is displayed as the library data object G23 in the library binder window G22 by S216.

[0151]When the <search of library> application 26 uses the library data 28 which a desktop management system manages, the publicly known arbitrary means of communication between programs are already used, and a message is sent to a desktop management system. The command of "a demand of library data" in this message, When library ID71 to demand and the library data 28 can be specified, the name of the information 28 to specify, for example, library data, and the information which specifies the data 18 which uses the library data 28 further are included.

[0152]The flow chart of drawing 60 explains the retrieval processing of the library data 28.

[0153]The desktop management system which received the message from the application 26, From the information which specifies the data 18 which analyzes a message and uses the library data 28 by S220. The object management information 110 on a library binder that the library data 28 of library ID71 demanded from the library information 70 which acquired the library information 70 of the room 13 to which the data belongs, and was before acquired by S221 is managed is acquired.

[0154]Whether the information which specifies the library data 28 by S222 is set up tests, and when set up, the library data 28 is searched with S223 from the object management information 110 on the obtained library binder 27. By S226, the obtained

library data 28 is notified to the application 26 using said means of communication.

[0155]When judged with on the other hand the information which specifies the library data 28 not being set up by S222, the application 26 is a case where the new library data 28 which a user chooses is required. In this case, a desktop management system is S224, opens the window of said library binder 27 from the object management information 110 on the obtained library binder 27, and displays the library data object G23 in the above-mentioned library binder 27. It waits for the selection input to the library data 28 by a user by S225. The library data 28 selected by S226 is notified to the application 26 using said means of communication.

[0156]When the application 26 while processing the <search of library by data request> data 18 requires the library data 28, It is searched from the library 17 of the room 13 where the data 18 used as the processing object of the application 26 exists, and when it does not exist there, it is searched from the library 17 of the office 14 where the room belongs. Namely, in the office 14 where the data 18 exists if the library data 28 is required from the application 26 while processing the data 18 of other offices 14, It refers to order called the library 17 of the room 13 of the owner of the data 18, and the library 17 of the office 14 where the above-mentioned owner's room 13 belongs. When the application 26 opens the data 18 copied to its room 13 from other users' room 13, It is searched by the library data 28 demand about the opened data 18 in order of the library 17 of its room 13, and the library 17 of the office 14 where the room is contained. It is also possible to copy automatically the library data 31 referred to when copying data to its room 13 from other rooms 13 to the library 17 of one's room 13. In this case, it is copied in the library binder 27 with the library ID71 [ same ] as the library data 28 copied. The retrieval order of the library data 28 of the data 18 can also specify the library 17 of the specific room 13, or the library 17 of the office 14 by obtaining office ID91 or the user ID 81 by the library search order information 150 shown in drawing 22. The library search order information 150 is information for every room 13, and a user can set up retrieval order arbitrarily.

[0157]In the example of drawing 22, although he is trying to set up to two, arbitrary numbers of retrieval order can be set up. When the library search order information 150 is not set up, it is searched in the above-mentioned order.

[0158]The <renewal of library data> library data 28 can update creation, edit, deletion, etc. completely like the usual data 18 with the application 26.

[0159]As are shown in drawing 6, and there is information on the number of times 74 of open which shows how many it is opened by the library data 28 in the library binder 27 in the library information 70 and it is shown in drawing 17 and drawing 18, The library update mode 84 and library update mode 95 which show whether it is opened by the library 17 of the room 13 and the office 14 by updating mode to User Information 80 and the office information 90, respectively. There is information on

(describing it as “library update mode” collectively hereafter), and renewal of the library data 28 can be managed.

[0160]It can restrict to the library 17 first opened by updating mode in the room 13 or the office 14, and can open by updating mode, and when opened [ above-mentioned ], “library update mode” is set up and renewal of the library data 28 is attained.

[0161]Also in the state where “library update mode” is set as the library 17 of the room 13 to which the data 18 exists when the application 26 requires the data 18, although opening of the data 18 is possible, “Reference is improper” returns to the response to the demand of the library data 28 from the application 26. That is, it is accepted library data 28 and the cotton of the data 18 which lacked is carried out to the application 26.

[0162]When the application 26 which set up “library update mode” closes the library data 28, setting out in “library update mode” is canceled and reference of the library 17 is attained. Update operation can also be forbidden for the library data 28 of the library binder 27 simultaneously opened also to the library data 28 of the library binder 27 opened only by reference mode. [ two or more ] If opened by how many library data 28, setting out of the number of times 74 of open considered as prohibition of update operation is arbitrary, and can be judged by the number of times 74 of open.

[0163]As shown in G21 of drawing 6, a desktop management system has a function which displays the number of times 74 of open of the library binder 27, and the user can know the number of times 74 of open.

[0164]The <clipboard> application 26 the data 18 or its part, The data which may copy or move to the inside of the application or one or more of other applications, therefore is created, and the place which stores the data temporarily are managed with the concept of clip data and a clipboard, respectively. This clipboard 24 has a window for displaying the stored clip data 32. The clip data 32 is managed like the object in the cabinet 16 or the binder 20 using the object management information shown by drawing 20. Format ID113 showing the data type is set to the clip data 32 by the application 26 which created the clip data 32.

[0165]The application 26 which requires the clip data 32 specifies the classification of the data demanded by format ID113 as a desktop management system. Next, it explains according to figures.

[0166]Like drawing 3 and drawing 11, the clipboard 24 which a desktop management system manages has a clipboard displayed in the user’s room window G01, and a clipboard displayed in the exit window G30. The clipboard expressed by the clipboard object G34 displayed on the exit window G30 is a clipboard attached to the office 14 to which the room 13 belongs.

The clipboard expressed by the clipboard object G071 displayed in a user’s room window G01 is a clipboard attached to the room 13.

Thus, the clipboard 24 is classified into the following two according to the gestalt of

the management.

[0167](1) The clipboard 24 attached to the room 13.

[0168]In the personal clipboard 24 registered as an object of the room 13, the useful range serves as one user. The useful range of the clipboard 24 can also be considered as two or more users' set specified arbitrarily by enabling it to access from other users by performing the same processing as open operation of the above-mentioned data 18.

[0169](2) The clipboard 24 attached to the office 14.

[0170]It is a user of the room 13 which is registered as an object of the office 14 and is contained in the clipboard 24 in the office 17, and the clipboard 24 which can be accessed in common, and the useful range serves as two or more sets everlastingly specified by the system.

[0171]The <registration to clipboard of clip data> application 26 explains operation and the procedure of registering the clip data 32 of a desktop management system, using the flow chart of drawing 62.

[0172]The application 26 already sends a message to a desktop using the publicly known arbitrary means of communication between programs. The command of "registering the clip data 32", and format ID113 of the clip data 32 to register are contained in this message. The desktop management system which received the message from the application 26 obtains format ID113 of the clip data 32 which analyzed the message by S230 and was specified by application.

[0173]Next, the message which stimulates attaching the name of the clip data 32 newly registered to a user by S231 is displayed, and the name of the clip data 32 is obtained.

[0174]Next, by S232, carry out copy processing of the data to a clipboard, and by S233. The clip data 32 is registered into the clipboard 24 with format ID113 specified as the obtained name by the application 26, and it displays as the clip data object G63 in the clipboard window G62 shown by drawing 61.

[0175]The <selection of clip data> application 26 explains the procedure for using the clip data 32 registered into the clipboard 24 of the desktop management system according to drawing 63.

[0176]When the application 26 uses the clip data 32 registered all over the clipboard 24 of a desktop management system, a message is already sent to a desktop management system using the publicly known arbitrary communications between programs. One or more available format ID113 are contained in this message with the command of "a demand of the clip data 32", and the application 26.

[0177]The desktop management system which received the message from the application 26 analyzes a message by S240, and obtains one or more format ID113 demanded.

[0178]Next, one or more clip data 32 with format ID which was specified by the



application 26 and which is in agreement with format ID113 is searched with S241 from inside out of the clipboard 24 to the clip data 32 registered into the clipboard 24. [0179]With next, the message to which it urges opening and displaying the clipboard window G62 by S242, and choosing the clip data 32 as a user. One or more clip data 32 with format ID which is in agreement with format ID113 which searched previously, and which was specified by the application 26 in the clipboard 24 is displayed as the clip data object G63.

[0180]By S243, if the clip data 32 is chosen by the user, a desktop management system will notify the selected clip data 32 to the application 26 S244 using said means of communication.

[0181]The function of a <cut buffer>, next the cut & copy & paste buffer (a "CCP buffer" is called hereafter) which a desktop management system has is explained.

[0182]The application 26 the data 18 or its part The inside of the application, Or the data which may copy or move to one or more of other applications, therefore is created, and the place which stores the data temporarily are managed with the concept of "CCP data" and a "CCP buffer", respectively. In the above-mentioned clipboard 24, these "CCP buffers" differs in a point without a means to display the stored "CCP data."

[0183]Drawing 64 is a figure showing the CCP data management information 160. The CCP data management information 160 is information for a desktop management system to manage "CCP data", and exists every room 13.

[0184]The following item about one "CCP data" is included in the CCP data management information 160. They are the data size 163 of format ID162 showing the data type of the file name 161 which is the contents of CCP data, and CCP data, and CCP data, the user ID 164 of the user who created CCP data, and the group ID 165 of the user who created CCP data. When one "CCP data" is stored in a "CCP buffer" for every format ID162, the application 26 specifies format ID162 and registration to a "CCP buffer" is performed, When the "CCP data" of format ID162 already specified as the "CCP buffer" is registered, only what performed registration afterwards is stored as effective "CCP data."

[0185]Application is specifying format ID162 and can obtain the "CCP data" of format ID162 of the specification which exists in a "CCP buffer." Although two or more clip data 32 was able to be stored in the above-mentioned clipboard 24 for every format ID113, When the application 26 specified format ID113 and requires the clip data 32, The clip data object G63 in which the clip data 32 of the format ID113 is shown needed to be displayed on the displayed clipboard window G62, and the user needed to be made to choose it. However, in this "CCP buffer", it differs from a clipboard that it is not necessary to make a user choose the "CCP data" of format ID specified by application like the clipboard 24, either, since only at most one exists. Although the "CCP data" stored in a "CCP buffer" is managed by the desktop management system,

unlike the data in the usual cabinet 16, a user cannot see it visually.

[0186]"CCP data" is created by the application 26 and format ID162 showing the classification of the data is set up by the application. The "CCP data" created by the application 26 is stored in the "CCP buffer" attached to the room of the user who is operating the application 26. Each "CCP data in a "CCP buffer"" has respectively peculiar format ID162. By requiring "CCP data" with format ID162 with the specific application 26 from a desktop management system, a desktop management system, "CCP data" with format ID162 in the "CCP buffer" attached to the user's room 13 can be specified, and application can be passed. Next, it explains according to figures.

[0187]The <registration to CCP buffer> application 26 explains the procedure of registering "CCP data" into the "CCP buffer" which a desktop management system manages, according to drawing 65.

[0188]The application 26 already sends a message to a desktop management system using the publicly known arbitrary means of communication between programs. The command of "registering CCP data", and format ID162 of the "CCP data" specified by the application to register are contained in this message.

[0189]The desktop management system which received the message from the application 26 analyzes a message by S250, and obtains format ID162 of the "CCP data" specified by the application 26.

[0190]Next, "CCP data" is registered into a "CCP buffer" with format ID162 specified by the application 26 by S251.

[0191]The <acquisition of CCP data> application 26 explains the procedure for using the "CCP data" registered into the "CCP buffer" which a desktop management system manages according to drawing 66.

[0192]When the application 26 uses the "CCP data" of the "CCP buffer" of a desktop management system, the publicly known arbitrary communications between programs are already used, and a message is sent to a desktop. The command of "a demand of CCP data" and format ID162 for specifying "CCP data" are contained in this message. The desktop management system which received the message from the application 26 analyzes a message by S260, and obtains format ID162 demanded.

[0193]Next, the "CCP data" which is in agreement with format ID162 received by S261 is searched. It is judged by S262 whether the "CCP data" which has specific format ID162 specified by the application 26 as a result of search exists. When "CCP data" with specific format ID162 specified by the application 26 is not found, it ends as it is.

[0194]When "CCP data" with specific format ID162 carried out by carrying out with the application 26 as a result of judgment of S262 is able to be obtained, the only "CCP data" in a "CCP buffer" is notified to the application 26 by S263 using said means of communication.

<Search> search makes a retrieval object the binder 20 under the cabinet 16, the

library binder 27, and the data 18, for example, sets up a search condition to these, and displays the binder 20, the library binder 27, and the data 18 corresponding to a search condition on a window.

[0195]A search condition is the information accompanying the binder 20, the library binder 27, and the data 18 which are retrieval objects, for example, In this example, they are each items, such as 118 the registered date 117 of the object management information 110 shown in drawing 20, and with an updated date, the Chinese character name 115, the owner user ID 120, the owner group ID 121, and the object type 111, and these logical combination. The method of a search condition may be a form which shows a user altogether the search condition which can be taken, for example as a choice, and is chosen from them, and may be the form of making a user inputting in detail. A way only a specific period until it starts (\*\*) retrieval processing and ends appropriates a specific region for a search-results display, and does not have a search-results displaying means for exclusive use about the method of showing a user search results, for example.

[0196](\*\*) A method with the search-results displaying means which newly generates the window for exclusive use for displaying only the thing corresponding to a search condition, etc.

[0197](\*\*) How to combine both (\*\*) and (\*\*) if needed. \*\*\*\* can be considered.

[0198]Next, the example of the method of showing a user retrieval processing and its search results is described.

[0199]Each of a method which shows a user the above-mentioned search results about the retrieval processing of the binder 20 which first exists in the hierarchy of cabinet 16 directly under is explained. Although how to use the object management information 110 on the cabinet 16 on the occasion of retrieval processing in this example is described, the example performed by creating the data structure for retrieval processings is also considered. Although simple conditions, such as the contents of the object type 111 of the object management information 110, are set up as search information in the expedient top of explanation, and this example, it does not adhere to setting out of a search condition as mentioned above.

[0200]The example of said (\*\*) is described using drawing 6, drawing 67, and flow chart drawing 68. The window which carries out the list display of the object in the cabinet 16 shall also be appropriated for a search-results display.

[0201]As the object in all the cabinets 16 shows drawing 5 at first, the list display is carried out into the cabinet window G16. If retrieval processing is performed as "searching the binder 20 in the cabinet 16" to these, It is inspected whether the object type 111 of the object management information 110 on the cabinet 16 is a binder as a search condition (S300), and it eliminates the display of what agree and is not from the cabinet window G16 (S301). This processing is performed to all the entries (object) of the cabinet 16, and the presentation to the user of search results

(S302) is completed.

[0202]As a result, as shown in drawing 67, in the cabinet window G100, a remains indication only of the binder object G101 which shows the binder 20 corresponding to a search condition is given, and what did not agree (namely, data 18) is non-display.

[0203]Next, the example of (\*\*) is described using drawing 5, drawing 69, and flow chart drawing 70. As the object in all the cabinets 16 shows drawing 5 at first, the list display is carried out into the cabinet window G16. If retrieval processing is performed as "searching the binder 20 in the cabinet 16" to these, The search-results viewing window G110 as shown in drawing 69 is generated (S310), It is inspected whether the object type 111 of the object management information 110 on the cabinet 16 is a binder as a search condition (S311), and it displays what agreeing on the search-results viewing window G110 as the search-results object G111 (S312). This processing is performed to all the entries (object) of the cabinet 16, and the presentation to the user of search results (S313) is completed.

[0204]As a result, although the display of the cabinet window G16 is eternal, as shown in drawing 69, only the binder 20 corresponding to conditions is newly displayed on the search-results viewing window G110 as the search-results object G111.

[0205]Next, the example of (\*\*) is described using drawing 5, drawing 67, drawing 71, and flow chart drawing 73.

[0206]As the object in all the cabinets 16 shows drawing 5 at first, the list display is carried out into the cabinet window G16. If retrieval processing is performed as "searching the binder 20 in the cabinet 16" to these, The search-results detailed window G120 as shown in drawing 71 is generated (S330), It inspects whether the object type 111 of the object management information 110 on the cabinet 16 is a binder as a search condition (S331), and the display of what agree and is not from the cabinet window G16 is eliminated (S332). The attendant information of the thing corresponding to conditions is displayed on the search-results detailed window G120 (S333). This processing is performed to all the entries (object) of the cabinet 16, and the presentation to the user of search results (S334) is completed.

[0207]As a result, in the cabinet window G16, as shown in drawing 67, a remains indication only of the binder object G17 which shows the binder 20 corresponding to a search condition is given, and what did not agree (namely, data 18) is non-display. As shown in drawing 71, the detailed information accompanying the binder 20 by which it was indicated by remains is displayed on the cabinet window G16 by the search-results detailed window G120.

[0208]With the almost same procedure as the retrieval processing procedure of the binder 20 which exists in the hierarchy of cabinet 16 directly under who stated above. Search of the data 18 which the hierarchy of cabinet 16 directly under has, search of the library binder 27 which the hierarchy of cabinet 16 directly under has, Search of the library data 28 which the hierarchy of cabinet 16 directly under has, search of the

binder 20 which the hierarchy of binder 20 directly under has, Search of the data 18 which the hierarchy of binder 20 directly under has, search of the library binder 27 which the hierarchy of library binder 27 directly under has, Search of the data 28 which the hierarchy of library binder 27 directly under has, search of the open tray 29 which the hierarchy of open shelf 22 directly under has, Search etc. of the data 18, the binder 20, etc. which the hierarchy of tag tray 19 directly under has can be realized by setting up the object type 111 as a search condition.

[0209]The example of the following retrieval processings explains according to the example which generates the search-results window G110 of explanation for exclusive use which displays the search results of (\*\*) for convenience.

[0210]Next, it explains, using flow chart drawing 72 about the retrieval processing of the object which exists under 2 hierarchies from the cabinet 16.

[0211]The object which exists under 2 hierarchies of the cabinet 16 refers to the object which the binder 20 exists, for example in the direct lower floor layer of the cabinet 16, and exists in the direct lower floor layer of the binder 20, and it says. If "the data which exists under 2 hierarchies from the cabinet 16" is set up as a search condition and search is carried out, the search-results window G110 for displaying search results will be generated (S320). The object type 111 chooses the entry which is a binder from the object management information 110 on the cabinet 16 (S321).

[0212]Read the object management information 110 based on the information on this entry, and it is inspected whether the object type 111 is the data 18 (S322), Read icon files from format ID113 of the agreeing entry, and the data information 50, and it displays on the search-results window G110 as the search-results object G111 (S324), This processing is performed to all the entries of the object management information 110 on this binder 20 (S325), and it returns to the processing (S321) as which the object type 111 chooses the entry which is the binder 20 from the object management information 110 on the cabinet 16.

[0213]This processing is performed to all the entries of the object management information 110 on the cabinet 16, and retrieval processing (S326) is ended.

[0214]It can search irrespective of a hierarchy number by performing such processing recursively. When a search condition is set up by having made into the retrieval object the object which exists in a different hierarchy and search is carried out, it can realize by changing the recursive number of execution time of retrieval processing. This is explained using flow chart drawing 74. Irrespective of a hierarchy, the object which the lower part hierarchy of the cabinet 16 has shall be made into a retrieval object, and it shall refer to this example by making into a search condition what "the object type 111 of the object management information 110 is data, and creator ID114 is 012." In this example, that the expedient upper data 18 of explanation may exist presupposes that it is directly under [ cabinet 16 ] or directly under binder 20.

[0215]If a search condition is set up by making into a retrieval object the object which

exists in an arbitrary hierarchy lower part from the cabinet 16 and search is carried out, the search-results window G110 for displaying search results will be generated (S340). Next, by making the cabinet 16 into a retrieval object, a search condition is passed, and "processing" which is closed processing is performed and it ends (S341). [0216]Next, "processing" of Step S341 of drawing 74 is explained using flow chart drawing 75. "Processing" like not only the cabinet 16 but the binder 20, the library binder 27, the open shelf 22, the open tray 29, and the tag tray 19, It is a module which realizes the function which displays the information on the data 18 which carries out search extraction of the data 18 recursively, and agrees in a search condition to the object which can have the further lower part hierarchy on the search-results window G110.

[0217]If "processing" is called, the object type 111 will choose the entry which is the data 18 from the object management information 110 on the object specified at the time of a call (S350). It is judged whether it agrees that the object type 111 of the search condition "object management information 110 is data to the entry, and creator ID114 is 012" (S351). Icon files are read from format ID113 of the agreeing entry, and the data information 50, and it displays as the search-results object G111 with information required for the search-results window G110 (S352).

[0218]On the other hand, the object type 111 calls "processing" with a search condition and a retrieval object recursively to the entry which is a binder from the object management information 110 on the object specified at the time of a call (S354). These processings are performed to all the entries of the object management information 110 on the object specified at the time of a call (S353), and a return is carried out from "processing."

[0219]Based on <attainment to search results>, next search results, the processing which reaches the object corresponding to a search condition is explained, without following the managing structure of data. In this example, it is taken as the data 18 under the cabinet 16, and the binder 20 that it is a retrieval object. First, the processing which generates the information for reaching the object corresponding to a search condition is explained using drawing 79, without following data management structure according to search results.

[0220]If retrieval processing is performed, as an object which generates the search-results window G110 (S360) and to which this search-results window G110 has a layered structure in that low rank like the cabinet 16 or the binder 20 simultaneously, The object management information 110 which stores the information about the search-results object in the search-results window G110 is generated (S361). What (S362) agreed in the search condition with execution of retrieval processing is displayed on the search-results window G110 (S363). The entry information to which the object management information 110 on the retrieval object cabinet 16 used in order to judge agreeing in a search condition simultaneously

corresponds is copied to the entry of the object management information on the search-results window G110 (S364). This processing is performed about all the retrieval objects (S365).

[0221]In this way, the processing at the time of open operation being performed to the constituted search-results window G110 is explained using drawing 77. The search-results object G111 which shows the binder 20 and the data 18 corresponding to a search condition is displayed on the search-results window G110.

The processing as the operation which shows the binder 20 and the data 18 which are substance of respectively as opposed to the binder object G17 and the data object G18 that the operation to the search-results object G111 in the search-results window G110 is the same is performed.

[0222]Namely, if open operation is performed to the search-results object G111 on the search-results viewing window G110, The information on the entry applicable to the specified object is acquired from the data management information 110 of the search-results window G110 (S370), and the open shop operation of the binder (S371) 22 or the data 18 is called according to the classification of the specified object (S372, S373). About the deleting operation to search results, forbidding this The search-results window G110 top, Only the attainment index to the data 18 or the binder 20 which a search-results object shows is deleted, and the binder 20 or the data 18 which is substance can also be set up not delete.

[0223]Like <presentation of the search results to the user in consideration of a layered structure> next the data 18, and the binder 20, The example of the search-results presentation to the user in consideration of the layered structure at the time of the object which makes a layered structure agreeing in a search condition simultaneously is described using flow chart drawing 78, drawing 2, drawing 79, and drawing 80. In this example, a search condition is set up by making into a retrieval object the data 18 and the binder 20 which exist in an arbitrary hierarchy lower part from the cabinet 16, It is assumed that the data 18 and the binder 20 in a layered structure as already explained retrieval processing carried out and shown in drawing 79 (A) with a thick frame as a result agreed in the search condition. Search results assume that it is displayed on the search-results window G110 as a search-results object like drawing 79 (B) regardless of the layered structure of actual data.

[0224]If a name opens here the search-results object which shows the binder 20 which is "the binder 0" from the search-results window G110, As already explained, the operation to a search-results object is opened by the binder 20 as operation to the binder 20 which it shows, and the binder window G22 is displayed. In the display of this binder window G22, The binder object G20 or the data object G18 in which the binder 20 which the search-results object G111 displayed in the search-results window G110 shows, or the data 18 is shown, Since it specifies that it is the object

which corresponded to the search condition by previous search, the display of an icon being adding shading like drawing 80 is changed. The processing for performing such a display is explained using flow chart drawing 78.

[0225]If the search-results object G130 which shows the binder 20 of the name "binder 0" of drawing 79 is clicked two and it is opened, The binder window G22 is generated (S380), Icon files are read from format ID113 of each entry of the object management information 110 on the binder 20 of the search-results object G130, and the data information 36, and an object is displayed with information required for said window (S381). Next, it is investigated whether the same information as this entry is included in the entry of the object management information 110 on the search-results window G110 (S382). When contained, it carries out by superimposing shading processing as shown, for example in drawing 80 on the icon display to said window, and a user's discernment is made easy (S383). The above processing is performed to all the entries of the object management information 110 on the binder 20 of a name "binder 0", and it ends (S384).

[0226]Next, an application program is started to search results and the example which performs arbitrary processings is described using flow chart drawing 81 and drawing 82. Drawing 82 is the search-results window G110.

The search-results object G111 corresponding to a search condition shown data 18 is displayed.

The processing performed to these search results shall start and carry out the printing job of the application program which performs printing in this example, although many edits, printings, etc. are considered. Starting of the application program which processes is performed in the form which chooses processing from a list, for example.

[0227]the search-results object G150 which shows the data 18 is chosen, and the manual operation button G140 on the search-results window G110 chooses -- having (S390). OFF ID51 and creator ID114 of the data 18 which are chosen are obtained from the entry of the data 18 applicable to the search-results object G150 as which the object management information 110 on the search-results window G110 was chosen (S391). This is compared with the application information 60, the processing name of the processing list 65 applicable to the format ID51 in an entry with the application ID61 [ same ] as creator ID114 is obtained (S392), and it displays as the menu list G141 (S393).

[0228]A user's selection of one item of the list G141 will obtain the file name 112 from the entry of the object management information 110 on the search-results viewing window G110 (S395). (S394) This file name 112, format ID51 which were already obtained, and the process code corresponding to the item chosen from the list G141 are passed to an application program, Application is started by the execution module name 62 corresponding to application ID61 (S396). When two or more data is chosen,



it can carry out by repeating a series of above-mentioned processings.

[0229]Processing when <execution of the retrieval processing in consideration of public information>, next a search to the object which other users own are performed is explained using flow chart drawing 83. In this example, the user A shall make a retrieval object the 16 or less cabinet of the user B data 18, and the 16 or less cabinet [ of the user C ] data 18, shall set up a search condition, and shall perform retrieval processing. The user's B cabinet 16 is opened to the user A who is an execution user of retrieval processing by the method according to the already explained open operation (this state). it claims "the user A is the user's B open user about this object" -- no cabinet 16 of the user C shall be exhibited to the user A (this state is called "the user A is the user's C secret user about this object")

[0230]If the user A sets up a search condition by making the users' B and C cabinet 16 into a retrieval object and performs retrieval processing, the search-results viewing window G110 will be generated (S400). When searching other users' cabinet 16, etc., The entry which corresponds to the cabinet 16 which it is going to search from the object management information 110 on the open tray 29 is looked for (S401), and when it exists, "processing" shown in flow chart drawing 75 is called with a search condition to this cabinet 16 (S402). When an applicable entry does not exist, it indicates that the object which the user specified as a retrieval object is not opened to the user A (S403). The retrieval processing to the object in which other users own this repeatedly (S404) to all the objects which the other users specified as this processing retrieval object own is ended.

[0231]Next, when a search to the object which other users own is performed, retrieval processing is explained using flow chart drawing 84 and drawing 85 about the processing which deters presentation of the search results to a user according to public information, after performing. In this example, the user A shall make a retrieval object the 16 or less cabinet of the user B data 18, and the 16 or less cabinet [ of the user C ] data 18, shall set up a search condition, and shall perform retrieval processing. The user's B cabinet 16 shall be opened to the user A who is an execution user of retrieval processing by the method according to <open operation>, and no cabinet 16 of the user C shall be exhibited to the user A.

[0232]If the user A sets up a search condition by making the users' B and C cabinet 16 into a retrieval object and performs retrieval processing, the search-results viewing window G110 will be generated (S410).

[0233]Next, "processing without a display" mentioned later is called with a search condition to the specified cabinet 16 (S411). Next, the entry which corresponds to the searched cabinet 16 from the object management information 110 on the open tray 29 is looked for (S412), and when it exists, search results are displayed on the search-results viewing window G110 based on the information which "the processing without a display" returned (S413). When an applicable entry does not exist, from the

information which "the processing without a display" returned. The information which is not pointed out and supported even if known by the secret user-group user, the information the necessity of getting to know for the reasons of management of a system, etc. even if it is a secret user-group user is accepted to be, etc. are selectively displayed on the search-results viewing window G110 (S414). The retrieval processing to the object in which other users own this processing repeatedly (S415) to all the objects which the other users specified as a retrieval object own is ended.

[0234]the information which is not pointed out and supported even if known by the aforementioned secret user is existence (/it exists -- don't carry out) of the object which agrees, for example in a search condition etc.

[0235]The information the necessity of getting to know for the reasons of management of a system, etc. is accepted to be even if it is said secret user is the size of an object displayed without displaying the information which can specify objects, such as a name of an object, a registered date, with an updated date, etc.

[0236]Next, "processing without a display" of Step S411 of drawing 84 is explained using flow chart drawing 85. "Processing without a display" like not only the cabinet 16 but the binder 20, the library binder 27, the open shelf 22, the open tray 29, and the tag tray 19. It is a module which realizes the function which displays the information on the data 18 which carries out search extraction of the data 18 recursively, and agrees in a search condition to the object which can have the further lower part hierarchy on the search-results viewing window G110.

[0237]"display -- if processing" which is not twisted is called, a return information area will be secured and initialized (S420), and the object type 111 will choose the entry which is the data 18 from the object management information 110 on the object specified at the time of a call (S421). The information on the entry which judges and (S422) agrees [ whether it agrees with a search condition to the entry and ] is set as return information (S423). On the other hand, the object type 111 calls "processing without a display" with a search condition and a retrieval object recursively to the entry which it is except data 18 from the object management information 110 on the object specified at the time of a call (S424). These processings are performed to all the entries of the object management information 110 on the object specified at the time of a call (S425), and the return of the return information is carried out from "processing without a display."

[0238]Processing when a search to the object which <execution of the retrieval processing in consideration of the public information to the user group>, next other users own is performed is explained using flow chart drawing 86. In this example, the user A shall make a retrieval object the 16 or less cabinet of the user B data 18, and the 16 or less cabinet [ of the user C ] data 18, shall set up a search condition, and shall perform retrieval processing. The user's B cabinet 16 is opened to the user group to which the user A who is an execution user of retrieval processing belongs by the

method according to <open operation> (this state). . Claim "The user A is the user's B open user-group user about this object." No cabinet 16 of the user C shall be exhibited to the user group to which the user A belongs (this state is called "the user A is the user's C secret user-group user about this object").

[0239]If the user A sets up a search condition by making the users' B and C cabinet 16 into a retrieval object and performs retrieval processing, the search-results viewing window G110 will be generated (S430). When searching other users' cabinet 16, etc., The entry which corresponds to the cabinet 16 which it is going to search from the object management information 110 on the open tray 29 is looked for (S431), and when it exists, "processing" shown in flow chart drawing 75 is called with a search condition to this cabinet 16 (S432). When an applicable entry does not exist, the object which the user A specified as a retrieval object indicates not opened to the user group to which the user A belongs (S433). The retrieval processing to the object in which other users own this processing repeatedly (S434) to all the objects which the other users specified as a retrieval object own is ended.

[0240]Next, when a search to the object which other users own is performed, retrieval processing is explained using flow chart drawing 87 and drawing 2 about the processing which deters presentation of the search results to a user according to public information, after performing.

[0241]In this example, the user A shall make a retrieval object the 16 or less cabinet of the user B data 18, and the 16 or less cabinet [ of the user C ] data 18, shall set up a search condition, and shall perform retrieval processing. The user's B cabinet 16 is opened to the user group to which the user A who is an execution user of retrieval processing belongs by the method according to <open operation>, The user's C cabinet 16 shall be opened to the user group to which the user A belongs, and no cabinet 16 of the user C shall be exhibited to the user group to which the user A belongs.

[0242]If the user A sets up a search condition by making the users' B and C cabinet 16 into a retrieval object and performs retrieval processing, the search-results viewing window G110 will be generated (S440). Next, "processing without a display" of drawing 85 described by <execution of the retrieval processing in consideration of public information> is called with a search condition to the specified cabinet 16 (S441). the information which looked for the entry which corresponds to the searched cabinet 16 from the object management information 110 on the open tray 29 (S442), and was returned in "processing without a display" next when it existed -- origin -- the search-results viewing window G110 -- \*\*\*\*\* (S443) of search results. The information which is not pointed out and supported even if known by the secret user from the information which "the processing without a display" returned when an applicable entry does not exist, Even if it is a secret user, the information etc. the necessity of getting to know for the reasons of management of a system, etc. is

accepted to be are selectively displayed on the search-results viewing window G110 (S444). The retrieval processing to the object which owns this processing in other users repeatedly (S445) to all the objects which the other users specified as a retrieval object own is ended.

[0243]By giving an indication which connected the icon which shows the application which can process a user's specified data with highlighting and a line, as explained above, By processing in which it keeps away or it brings the distance of both icons close. By showing a user easily the correspondence relation of the user data which can be referred to and updated with application, and providing the starting method of easy application, Even if it does not learn the starting method of application, without a user being conscious of matching of data and application, reference of data and updating are attained.

[0244]It is possible on two or more computer systems with two or more users to search simply, without arbitrary users being conscious of the structure and the computer system with which data is managed in the arbitrary data currently distributed, and it is easy to process the searched data.

[0245]one arbitrary users' retrieval processing to the arbitrary data currently distributed on two or more computer systems with two or more users -- or it becomes possible to perform restrictively the display as a result of retrieval processing. It becomes possible by furthermore having a user management tool, the second operation, a management tool, and the second retrieval execution control means to perform restrictively one user's retrieval processing to the arbitrary data currently distributed, or the display of a retrieval processing result on two or more computer systems with two or more users.

[0246]Even if it applies this invention to the system which comprises two or more apparatus, it may be applied to the device which comprises one apparatus. It cannot be overemphasized that this invention can be applied also when attained by supplying a program to a system or a device.

[0247]

[Effect of the Invention]As explained above, desired data can be operated without according to the data processing device concerning this invention, a user being conscious of it, wherever data may be stored in. It is effective in that a user does not need to be conscious of correspondence with data and the application program which can process it.

---

## DESCRIPTION OF DRAWINGS

---

[Brief Description of the Drawings]

[Drawing 1] It is a block diagram of the data processing device of an example.

[Drawing 2] It is an explanatory view of a room and an office in a desktop management system.

[Drawing 3] It is a display example of a user's room window.

[Drawing 4] It is a display example of a desktop window.

[Drawing 5] It is a display example of a cabinet window.

[Drawing 6] It is a display example of a library window.

[Drawing 7] It is a display example of a library binder window.

[Drawing 8] It is a display example of a tag tray window.

[Drawing 9] It is a display example of an open shelf window.

[Drawing 10] It is a display example of an open window.

[Drawing 11] It is a display example of an exit window.

[Drawing 12] It is a display example of the exit included in an office.

[Drawing 13] It is a display example of the exit included in a room.

[Drawing 14] It is an example of data information.

[Drawing 15] It is an example of application information.

[Drawing 16] It is an example of library information.

[Drawing 17] It is an example of User Information.

[Drawing 18] It is an example of office information.

[Drawing 19] It is an example of group information.

[Drawing 20] It is an example of object management information.

[Drawing 21] It is an example of open object management information.

[Drawing 22] It is an example of library search order information.

[Drawing 23] They are operation instances which open the binder in a cabinet.

[Drawing 24] It is a flow chart of the processing which opens a binder.

[Drawing 25] They are operation instances of the data copy between a cabinet and a binder.

[Drawing 26] It is a flow chart of the data copy processing between a cabinet and a binder.

[Drawing 27] They are operation instances which open an application object.

[Drawing 28] It is a flow chart of the starting processing of an application program.

[Drawing 29] They are operation instances which open an application object.

[Drawing 30] It is a flow chart of the starting processing of an application program.

[Drawing 31] They are operation instances which open an application object.

[Drawing 32] They are operation instances which open a data object.

[Drawing 33] It is a flow chart of the starting processing of an application program.

[Drawing 34] They are operation instances which open a data object.

[Drawing 35] It is a flow chart of the starting processing of an application program.

[Drawing 36] It is an example of copy operation of the application object between a cabinet and a desktop.

[Drawing 37]It is a flow chart of application object moving processing.

[Drawing 38]They are drag operation instances of the data object from a cabinet to a desktop.

[Drawing 39]It is the example which tied and displayed the application object and the data object.

[Drawing 40]It is a flow chart of the processing which ties and displays an application object and a data object.

[Drawing 41]It is the example which displayed the relation between an application object and a data object in the highlight.

[Drawing 42]It is a flow chart of the processing which displays the relation between an application object and a data object in a highlight.

[Drawing 43]It is a display example of guidance by attraction.

[Drawing 44]It is a flow chart of processing which starts application by guidance by attraction.

[Drawing 45]It is a display example of reaction force \*\*\*\* guidance.

[Drawing 46]They are open tray creation operation instances.

[Drawing 47]It is an example of open setting operation.

[Drawing 48]It is a flow chart of open tray creation processing.

[Drawing 49]It is a flow chart of open operation processing.

[Drawing 50]It is a flow chart of open setting processing.

[Drawing 51]It is a flow chart of exit open shop operation.

[Drawing 52]It is a flow chart of tag creation processing.

[Drawing 53]It is a flow chart of tag open shop operation.

[Drawing 54]It is a display example of data abandonment operation.

[Drawing 55]It is a flow chart of data discarding treatment.

[Drawing 56]It is a flow chart of processing of a data abandonment supervisory process.

[Drawing 57]It is an example of library binder creation operation.

[Drawing 58]It is a flow chart of library binder processing.

[Drawing 59]It is a flow chart of library data registration processing.

[Drawing 60]It is a flow chart of library data retrieval processing.

[Drawing 61]It is a display example of a clipboard window.

[Drawing 62]It is a flow chart of clip data registration processing.

[Drawing 63]It is a flow chart of clip data retrieval processing.

[Drawing 64]It is an example of CCP data management information.

[Drawing 65]It is a flow chart of the data registration processing to a CCP buffer.

[Drawing 66]It is a flow chart of data retrieval processing of a CCP buffer.

[Drawing 67]It is a display example of search results.

[Drawing 68]It is a flow chart of retrieval processing.

[Drawing 69]It is a display example of search results.

[Drawing 70] It is a flow chart of retrieval processing.

[Drawing 71] It is a detailed display example of search results.

[Drawing 72] It is a flow chart of retrieval processing.

[Drawing 73] It is a flow chart of retrieval processing.

[Drawing 74] It is a flow chart of the retrieval processing independent of a hierarchy.

[Drawing 75] It is a flow chart of the retrieval processing independent of a hierarchy.

[Drawing 76] It is a flow chart of retrieval processing.

[Drawing 77] It is a flow chart of the open shop operation of a search-results window.

[Drawing 78] It is a flow chart of display processing of search results.

[Drawing 79] It is an example for retrieving operation.

[Drawing 80] It is a display example of search results.

[Drawing 81] It is a flow chart of processing which starts application about search results.

[Drawing 82] It is an example of operation which starts application about search results.

[Drawing 83] It is a flow chart of retrieval processing.

[Drawing 84] It is a flow chart of display processing of the search results in consideration of public information.

[Drawing 85] It is a flow chart of display processing of the search results in consideration of public information.

[Drawing 86] It is a flow chart of the retrieval processing in consideration of public information.

[Drawing 87] It is a flow chart of display processing of the search results in consideration of public information.

[Drawing 88] It is a flow chart of processing which starts application by guidance by reaction force.

[Description of Notations]

- 1 -- CRT,
- 2 -- Video RAM (VRAM),
- 3 -- Bit processing unit (BMU),
- 4 -- Keyboard,
- 5 -- Pointing device (PD),
- 6 -- CPU,
- 7 -- ROM,
- 8 -- RAM,
- 9 -- Hard disk drive (HDD),
- 10 -- Floppy disk drive (FDD),
- 11 -- Network interface (Net-I/F),
- 12 -- I/O bus
- 13 -- Room,

14 -- Office,  
15 -- Desktop,  
16 -- Cabinet,  
17 -- Library,  
18 -- Data,  
19 -- Tag tray,  
20 -- Binder,  
21 -- Tag,  
22 -- Open shelf,  
23 -- Garbage can  
24 -- Clipboard,  
25 -- Exit,  
26 -- Application,  
27 -- Library binder,  
28 -- Library data,  
29 -- Open tray,  
30 -- Open object,  
31 -- Door,  
32 -- Clip data,  
50 -- Data information  
60 -- Application information  
70 -- Library information  
80 -- User Information  
90 -- Office information,  
100 -- Group information  
110 -- It is object management information.



(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-131145

(43)公開日 平成6年(1994)5月13日

(51)Int.Cl. <sup>5</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 3/14	3 7 0 A	7165-5B		
12/00	5 1 5 B	8526-5B		
15/00	3 9 0	7459-5L		

審査請求 未請求 請求項の数13(全 60 頁)

(21)出願番号 特願平4-131137

(22)出願日 平成4年(1992)5月22日

(71)出願人 000001007

キャノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 渡邊 哲史

東京都大田区下丸子3丁目30番2号 キャ  
ノン株式会社内

(72)発明者 吉川 直廣

東京都大田区下丸子3丁目30番2号 キャ  
ノン株式会社内

(72)発明者 正田 芳郎

東京都大田区下丸子3丁目30番2号 キャ  
ノン株式会社内

(74)代理人 弁理士 大塚 康德 (外1名)

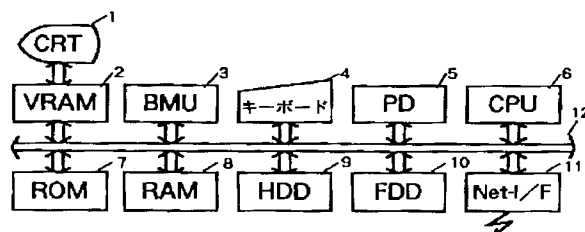
最終頁に続く

(54)【発明の名称】 データ処理装置

(57)【要約】

【目的】 利用者が、処理しようとするデータとそれを処理するプログラムの細部に関する知識を持たなくとも、処理を行えるデータ処理装置を提供する。

【構成】 HDD 9には階層構造で管理されているデータが格納されている。また、ネットワークインターフェース11は他のコンピュータシステムに接続されている。HDD 9にデータとともに、そのデータの属性等や、データ管理の単位ごとの属性、データとそれを処理できるプログラムの対応といった情報が格納されている。データ及びプログラムをアイコンで表示する場合、前記情報を基に処理可能な対を識別できる様にCRT 1に表示する。また、データの検索の際も、前記情報にデータの格納場所等を得るため、利用者に負担をかけることはない。



## 【特許請求の範囲】

【請求項1】 内蔵したアプリケーションプログラムを複数の利用者によって利用され、階層的な管理構造の下で利用者単位でデータを管理するデータ処理装置であって、

前記該アプリケーションプログラムで実行可能な処理の種類と、該処理の対象となる前記利用者単位で管理されたデータとの対応づけを管理する管理手段と、前記アプリケーションプログラムによる処理と前記データとをそれぞれ固有のアイコンで表現する手段と、

前記管理手段による前記アプリケーションプログラムでの処理と該プログラムの処理対象のデータとの対応を、それぞれに固有なアイコンにより表示する表示手段と、を備えることを特徴とするデータ処理装置。

【請求項2】 前記表示手段は、前記対応するアプリケーションプログラムでの処理のアイコンと該プログラムの処理対象のデータのアイコンをハイライト表示することを特徴とする請求項1記載のデータ処理装置。

【請求項3】 前記表示手段は、前記対応するアプリケーションプログラムでの処理と該プログラムの処理対象のデータのアイコンとともに該アイコンを結ぶ線を表示することを特徴とする請求項1記載のデータ処理装置。

【請求項4】 前記表示手段は、前記対応するアプリケーションプログラムでの処理のアイコンと該処理対象のデータのアイコンとが所定の距離以内に近づいた場合に、重ねあわせて表示し直すことを特徴とする請求項1記載のデータ処理装置。

【請求項5】 前記表示手段は、前記対応するアプリケーションプログラムでの処理のアイコンと該処理対象以外のデータのアイコンとが所定の距離以内に近づいた場合に、所定距離以上離して表示し直すことを特徴とする請求項1記載のデータ処理装置。

【請求項6】 階層的な管理構造の下で利用者単位でデータを管理するデータ処理装置であって、データの属性を管理する管理手段と、検索のための検索条件を設定する検索条件設定手段と、注目階層のデータの下にあって所望の階層数にわたるデータから、前記検索条件により設定された条件に合致するデータを検索する検索手段と、前記検索手段による検索の結果を利用者に提示する検索結果提示手段と、を備えることを特徴とするデータ処理装置。

【請求項7】 前記検索手段により得られた、検索条件に合致するデータを、該データの属性とともに注目階層に複写する複写手段を更に備え、階層的な構造をたどることなく、検索条件に合致したデータを参照できることを特徴とする請求項6記載のデータ処理装置。

【請求項8】 データ処理を行うプログラムを起動する手段を更に備え、前記検索手段により検索されたデータを対象としてデータ処理を行うことを特徴とする請求項

7記載のデータ処理装置。

【請求項9】 階層的な管理構造の下で利用者単位でデータを管理するデータ処理装置であって、前記データの階層的な管理構造を表示する構造表示手段と、

データを検索するための条件を設定する検索条件設定手段と、

注目階層のデータの下にあって所望の階層数にわたるデータから、前記検索条件設定手段により設定された条件に合致するデータを検索する検索手段と、

該検索手段による検索結果を利用者に提示する検索結果提示手段と、

前記検索により得られたデータの下にある下位データをその管理構造に従って表示する場合に、前記検索条件に合致する前記下位データを特別に表示する表示手段と、を備えることを特徴とするデータ処理装置。

【請求項10】 階層的な管理構造の下で利用者ごとにデータを管理するデータ処理装置であって、

前記データを操作する操作手段と、

前記データの操作を許された利用者の情報を管理する管理手段と、

データを検索する条件を設定する検索条件設定手段と、

注目階層のデータの下にあって所望の階層数にわたるデータから、前記検索条件により設定された条件に合致するデータを検索する検索手段と、

前記管理手段による利用者の情報に基づいて、前記検索手段による検索の実行を制御する検索実行制御手段と、を備えることを特徴とするデータ処理装置。

【請求項11】 前記操作手段によるデータの操作を許された利用者を利用者の集合として管理する利用者管理手段と、

前記利用者管理手段による利用者の集合の情報に基づいて、前記検索手段による検索処理の実行を制御する第二の検索実行制御手段と、を更に有することを特徴とする請求項10記載のデータ処理装置。

【請求項12】 階層的な管理構造の下で利用者ごとにデータを管理するデータ処理装置であって、

前記データを操作する操作手段と、

前記データの操作を許された利用者の情報を管理する管理手段と、

データを検索する条件を設定する検索条件設定手段と、注目階層のデータの下にあって所望の階層数にわたるデータから、前記検索条件により設定された条件に合致するデータを検索する検索手段と、

前記管理手段による利用者の情報に基づいて、前記検索手段による検索処理の結果を表示する表示手段と、を備えることを特徴とするデータ処理装置。

【請求項13】 前記操作手段によるデータの操作を許された利用者を利用者の集合として管理する利用者管理手段と、

前記利用者管理手段による利用者の集合の情報に基づいて、前記検索手段による検索処理の結果を表示する第二の表示手段と、を更に有することを特徴とする請求項1記載のデータ処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、例えば、複数の利用者を持つ複数のデータ処理装置上で、利用者が所望のデータを操作できるデータ処理装置等に関する。

【0002】

【従来の技術】従来、複数の利用者を持つデータ処理装置では、利用者のデータはデータ処理装置により定められたメモリ等の所定場所に保存されている。一利用者が、特定の条件に合致するデータを操作しようとする場合には、そのデータ処理装置特有の手続きによってデータの操作を実現できた。例えば、データ操作をアプリケーションプログラムにより行おうとする場合、そのアプリケーションプログラム特有の手続きによって実現できた。また、操作の対象となるデータが、何らかの形で接続された他のデータ処理装置上に存在する場合には、対象となるデータが存在するデータ処理装置を指定するなどの定められた手続きをとることによって実現できた。

【0003】また、従来の複数の利用者により利用されるデータ処理装置では、各利用者のデータは所定の記憶場所に格納されており、所望の条件に合致するデータを操作しようとする場合には、記憶場所や条件の指定等、データ処理装置に固有の手続きで実現できた。

【0004】

【発明が解決しようとする課題】しかしながら、上記従来例では、利用者はデータ処理装置ごとに固有の専門的知識が必要とされ、かつ、計算機システムごとの複雑な操作を習得しなければならなかった。更に、他のデータ処理装置上に存在するデータを操作する場合は、どの処理装置上にそのデータが存在するのかを知らなければならなかった。例えば、アプリケーションによりデータ操作を行う場合、利用者は、アプリケーションプログラムごとに実行可能な処理等の専門知識が必要となり、かつ、参照あるいは更新することのできるデータの種類やどこに格納されているかといったことを予め知っておかなければならなかった。また、逆に、利用者は、データごとにそれを参照あるいは更新することのできるアプリケーションプログラムおよび処理の種類をあらかじめ知っておかなければならなかった。

【0005】本発明は上記従来例に鑑みて成されたもので、データがどこに格納されていても利用者がそれを意識することなく、所望のデータを操作することのできるデータ処理装置を提供することを目的とする。

【0006】また、データとそれを処理することのできるアプリケーションプログラムとの対応を利用者が意識しなくともよいデータ処理装置を提供することを目的と

する。

【0007】

【課題を解決するための手段】及び

【作用】上記目的を達成するために本発明のデータ処理装置は次のような構成からなる。

【0008】内蔵したアプリケーションプログラムを複数の利用者によって利用され、階層的な管理構造の下で利用者単位でデータを管理するデータ処理装置であって、前記該アプリケーションプログラムで実行可能な処理の種類と、該処理の対象となる前記利用者単位で管理されたデータとの対応づけを管理する管理手段と、前記アプリケーションプログラムによる処理と前記データとをそれぞれ固有のアイコンで表現する手段と、前記管理手段による前記アプリケーションプログラムでの処理と該プログラムの処理対象のデータとの対応を、それぞれに固有なアイコンにより表示する表示手段とを備える。

【0009】また、階層的な管理構造の下で利用者単位でデータを管理するデータ処理装置であって、データの属性を管理する管理手段と、検索のための検索条件を設定する検索条件設定手段と、注目階層のデータの下にあって所望の階層数にわたるデータから、前記検索条件により設定された条件に合致するデータを検索する検索手段と、前記検索手段による検索の結果を利用者に提示する検索結果提示手段とを備える。

【0010】また、階層的な管理構造の下で利用者単位でデータを管理するデータ処理装置であって、前記データの階層的な管理構造を表示する構造表示手段と、データを検索するための条件を設定する検索条件設定手段と、注目階層のデータの下にあって所望の階層数にわたるデータから、前記検索条件設定手段により設定された条件に合致するデータを検索する検索手段と、該検索手段による検索結果を利用者に提示する検索結果提示手段と、前記検索により得られたデータの下にある下位データをその管理構造に従って表示する場合に、前記検索条件に合致する前記下位データを特別に表示する表示手段とを備える。

【0011】また、階層的な管理構造の下で利用者ごとにデータを管理するデータ処理装置であって、前記データを操作する操作手段と、前記データの操作を許された利用者の情報を管理する管理手段と、データを検索する条件を設定する検索条件設定手段と、注目階層のデータの下にあって所望の階層数にわたるデータから、前記検索条件により設定された条件に合致するデータを検索する検索手段と、前記管理手段による利用者の情報に基づいて、前記検索手段による検索の実行を制御する検索実行制御手段とを備える。

【0012】また、階層的な管理構造の下で利用者ごとにデータを管理するデータ処理装置であって、前記データを操作する操作手段と、前記データの操作を許された利用者の情報を管理する管理手段と、データを検索する

条件を設定する検索条件設定手段と、注目階層のデータの下にあって所望の階層数にわたるデータから、前記検索条件により設定された条件に合致するデータを検索する検索手段と、前記管理手段による利用者の情報に基づいて、前記検索手段による検索処理の結果を表示する表示手段とを備える。

【0013】

【実施例】

＜装置の説明＞図1は実施例におけるデータ処理装置のブロック図である。図において、1はアプリケーションプログラムによって処理中のデータの情報や各種メッセージメニューなどを表示するCRTを示している。2はCRT1の画面に表示されるイメージを展開するビデオRAM（以下、VRAM）を示している。3はメモリ間あるいはメモリと各デバイス間のデータ転送をコントロールするビット処理ユニット（以下、BMU）を示し、4は文書編集などの処理を行なうための各種キーを備えたキーボードであり、5は画像上のアイコンなどを指し示すために用いられるポインティングデバイス（以下、PD）を示す。6は本装置全体の制御を司るCPUであり、7はCPU6の動作処理手順を実現するプログラム等を記憶しているROMである。なお、このROM7にはデータ処理に係るアプリケーションプログラムやエラー処理プログラムをはじめ、後述する図26などで示すフローチャートに係るプログラムも記憶されている。8は上述した各種プログラムをCPU5が実行する時にワークエリアやエラー処理時の一時退避エリア等として用いられるRAMを示している。9はハードディスクドライブ（以下、HDD）、10はフロッピーディスクドライブ（以下、FDD）をそれぞれ示し、それぞれのディスクには後述するアプリケーションプログラムやデータ、ライブラリなどの保存用に用いられる。11はネットワークインターフェイス（以下、Net I/F）を示し、他のデータ処理装置との間でネットワークを経由してデータ転送を行なうためにネットワーク上のデータ制御、診断を行なう。12は上述した各ユニット間を接続するI/Oバス（アドレスバス、データバスおよび制御バスからなる）である。

【0014】＜デスクトップ管理システムの場合＞本実施例において、データを処理するアプリケーションプログラム及びそれらのアプリケーションプログラムによって処理されるデータを管理し、ユーザに対してそのデータを操作する環境を与え、アプリケーションプログラムの実行や、実行されたアプリケーションプログラムにデータを与える等の処理を行なうプログラムを「デスクトップ管理システム」と呼ぶ。実際のデスクトップ管理システムの表示例としては、図3で示すようにウィンドウシステム上に複数のウィンドウで構築されたものであり、キーボード4およびポインティングデバイス5などからの入力によってデータを操作する。

【0015】次にデスクトップ管理システムにおける種々の概念について説明する。

【0016】図2は、デスクトップ管理システムにおけるユーザの作業環境の概念であるルーム13とオフィス14について表したものである。ルーム13とは、1人のユーザが所有する環境を提供するスペースを指す概念であり、オフィス14とは、それに含まれるルーム13の集合で構成されるスペースを指す概念であり、実社会におけるオフィスとそこで働く作業員個人の作業環境であるルームを模したものである。

【0017】例えば、1つの計算機の物理的ファイルシステムを1つのオフィス14として、ネットワークで結合された複数の計算機に分散したオフィス14毎に、それぞれ複数のルーム13が存在する。ルーム13に含まれる作業環境を提供する機能については、後述の操作オブジェクトの説明で言及する。1人のユーザは、適当なオフィス14に自分のルーム13を持ち、そのルーム13内でデスクトップ管理システムのサービスを受け、後述の操作オブジェクトを操作することにより、アプリケーションプログラムの起動やデータの処理を行なうことが可能となる。また、ユーザは自分のルーム13外の同じオフィス14の他のユーザのルーム13あるいは他のオフィス14のルーム13にアクセスすることができ、許された範囲内において、自分のルーム13内と同様の操作をすることができる。さらに、複数のルーム13に跨ってデータの移動、複写などの操作を行なうことも可能である。

【0018】操作オブジェクト（オブジェクト）とは、ユーザによるデスクトップ管理システムへのアクセス手段を提供するために、ウィンドウ上にアイコンとして表現されるデータおよびデータの集合、さらにデスクトップ管理システムが提供する機能を示すものである。

【0019】次に、図2とそれぞれの図を用いてユーザインターフェイスで提供される操作オブジェクトを説明する。図2では、ルーム「高田」に種々のオブジェクトが属しているが、ルーム「佐藤」あるいはルーム「渡辺」も同様である。

【0020】図3は、ウィンドウシステム上で実現されるユーザのルーム13を表すルームウィンドウG01を示したものである。図2及び図3を用いてルームウィンドウG01内にあるオブジェクトを説明する。

【0021】デスクトップオブジェクトG02によって示されるデスクトップ15は、デスクトップ管理システムの一機能であるアプリケーションプログラムを管理する機能を示すオブジェクトである。デスクトップG02のオープン操作により、後述のデスクトップ内オブジェクトを含むウィンドウが表示される。

【0022】キャビネットオブジェクトG03によって示されるキャビネット16は、アプリケーションプログラムによって作成されたデータを階層構造で管理する機

能のトップレベルのエントリを指すオブジェクトである。キャビネットオブジェクトG03のオープン操作により、後述のキャビネット内オブジェクトを含むウィンドウが表示される。

【0023】ライブラリオブジェクトG04によって示されるライブラリ17は、アプリケーションプログラムが、データを処理する際に参照する（共通な）補助データであるライブラリが格納されている場所のエントリを指すオブジェクトである。ライブラリオブジェクトG04のオープン操作により、後述のライブラリ内オブジェクトを含むウィンドウが表示される。

【0024】タグトレイオブジェクトG05によって示されるタグトレイ19は、キャビネットの階層構造をたどることなく、後述するバインダ20やデータ18にアクセスするための仮想的なエントリを表すオブジェクトである。タグ21は格納するオブジェクトを指す。タグトレイオブジェクトG05のオープン操作により、後述のタグトレイ内オブジェクトを含むウィンドウが表示される。

【0025】公開棚オブジェクトG06によって示される公開棚22とは、ユーザ自身のルーム13内のキャビネット16以下のデータ18やバインダ20を他のユーザが参照あるいは更新することを許可する（以下、公開操作と呼称する）ための操作を実現し、その情報を保存するオブジェクトを指す。公開棚オブジェクトG06のオープン操作により、後述の公開棚内オブジェクトを含むウィンドウが表示される。

【0026】ゴミ箱オブジェクトG07によって示されるゴミ箱23は、データの廃棄を管理するオブジェクトを指す。クリップボードオブジェクトG08によって示されるクリップボード24は、アプリケーション26が処理するデータ18またはその一部をアプリケーション26間で交換するための機能を提供するオブジェクトである。

【0027】出口オブジェクトG09によって示される出口25は、他のユーザのルーム13やオフィス14へのアクセス手段を表すオブジェクトである。出口オブジェクトG09のオープン操作により、後述の出口内オブジェクトを含むウィンドウが表示される。

【0028】なお、操作オブジェクト以外であるが、G10はユーザの名称および表示されているウィンドウの表題を表示するタイトル、G11はウィンドウ内の操作を指示するための操作ボタン、G12は操作のヘルプ内容を表示するヘルプボタン、G13はウィンドウ外にあるアイコンをスクロールさせて表示させるスクロールバーを表している。

【0029】図4はデスクトップG02がオープンされた状態のデスクトップウィンドウG14を示したものである。図4を用いてデスクトップウィンドウG14内の操作オブジェクトを説明する。

【0030】アプリケーションオブジェクトG15によって示されるアプリケーション26は、アプリケーションプログラムを操作（起動）するためのオブジェクトを指す。アプリケーションオブジェクトG15の操作は後述する。

【0031】図5はキャビネットオブジェクトG03がオープンされた状態のキャビネットウィンドウG16を示したものである。図5を用いてキャビネットウィンドウG16内の操作オブジェクトを説明する。

【0032】バインダオブジェクトG17によって示されるバインダ20は、データを階層構造で管理するキャビネット16下のノードを示すオブジェクトであり、キャビネット16およびバインダ20の下に任意に作成することができる。バインダオブジェクトG17のオープン操作により、そのバインダ20内のバインダ20およびデータ18を含むウィンドウが表示される。

【0033】データオブジェクトG18によって示されるデータ18は、アプリケーション26により作成されたデータ18を表すオブジェクトでありデータ18を操作したり、そのデータ18を処理するアプリケーションプログラムを起動したりするために使用する。データオブジェクトG18の操作は後述する。バインダ20内のオブジェクトも同様である。

【0034】図6はライブラリオブジェクトG04がオープンされた状態のライブラリウィンドウG19を示したものである。図6を用いてライブラリウィンドウG19内の操作オブジェクトを説明する。

【0035】ライブラリバインダオブジェクトG20によって示されるライブラリバインダ27は、後述するライブラリとして管理される同種のライブラリデータ28をまとめて格納するためのオブジェクトである。G21はライブラリバインダ27下のライブラリデータ28がアプリケーションプログラムによって現在参照されている回数を表す数値である。ライブラリバインダオブジェクトG20のオープン操作により、オープンされたライブラリバインダ27内のデータ18を含むウィンドウが表示される。

【0036】図7はライブラリバインダオブジェクトG20がオープンされた状態のライブラリバインダウィンドウG22を示したものである。図7を用いてライブラリバインダウィンドウG22内の操作オブジェクトを説明する。

【0037】ライブラリデータオブジェクトG23によって示されるライブラリデータ28とは、アプリケーションにより作成されたデータ18のうち、特に前述のライブラリとして管理されるものを指すオブジェクトである。ライブラリデータ28は同一のデータ形式のものが同じライブラリバインダ27に格納される。

【0038】図8はタグトレイオブジェクトG05がオープンされた状態のタグトレイウィンドウG24を示し

たものである。図8を用いてタグトレイウィンドウG 24内の操作オブジェクトを説明する。

【0039】タグオブジェクトG 25によって示されるタグ21は、バインダ20、データ18のキャビネット内オブジェクトをキャビネットの階層構造をたどることなくアクセスするために仮想的に表示されるオブジェクトである。タグ21に対しては、そのタグ21が指し示している実体のバインダ20、データ18オブジェクトと同様の操作が可能である。

【0040】図9は公開棚オブジェクトG 06がオープンされた状態の公開棚ウィンドウG 26を示したものである。図9を用いて公開棚ウィンドウG 26内の操作オブジェクトを説明する。

【0041】公開トレイオブジェクトG 27によって示される公開トレイ29は、ユーザが公開操作を行なう場合に、公開対象となる他のユーザまたは、ユーザの集合（以下、ユーザグループと呼称する）毎に作成されるオブジェクトを指す。公開トレイオブジェクトG 27のオープン操作により、公開オブジェクト30を含むウィンドウが表示される。

【0042】図10は公開トレイオブジェクトG 27がオープンされた状態の公開トレイウィンドウG 28を示したものである。図10を用いて公開トレイウィンドウG 28内の操作オブジェクトを説明する。公開オブジェクトG 29は、公開操作が行なわれたユーザのキャビネット16、バインダ20、データ18のオブジェクトを仮想的に表示するものである。公開オブジェクトG 29が存在する公開トレイ29の属性により、その公開オブジェクトG 29を公開する対象となるユーザあるいはユーザグループが決定される。公開操作については後述する。

【0043】図61はクリップボードオブジェクトG 08がオープンされた状態のクリップボードウィンドウG 62を示したものである。クリップボードウィンドウG 62内にはクリップデータ32を示す操作オブジェクトであるクリップデータオブジェクトG 63が含まれる。クリップデータ32とはアプリケーション26により作成されたデータ18あるいはその一部と、そのフォーマットIDを含むデータである。アプリケーション26によって生成されたクリップデータ32は、そのユーザのルーム13内のクリップボード24内に格納され、クリップボードウィンドウG 62内にクリップデータオブジェクトG 63表示される。クリップボード24の操作については後述する。

【0044】図11は出口オブジェクトG 09がオープンされた状態の出口ウィンドウG 30を示したものである。図11を用いて出口ウィンドウG 30内の操作オブジェクトを説明する。ドアオブジェクトG 31は、ユーザが属するオフィスに存在する他のユーザのルームへの入口を表すオブジェクトである。ドア31を示すドアオ

ブジェクトG 31にはユーザ名が表示されており、ドアオブジェクトG 31をオープン操作することによりそのドア31の示すユーザのルーム13のウィンドウが表示され、そのルーム13のユーザがユーザまたはユーザを含むユーザグループに対して公開操作を行なったキャビネット16、バインダ20、データ18を示すオブジェクトが表示される。ユーザはその公開されたオブジェクトを操作することにより、そのユーザのデータをアクセスすることができる。

【0045】オフィスオブジェクトG 32は、他のオフィスへの入口を表すオブジェクトである。オフィスオブジェクトG 32にはオフィス名が表示されており、そのオープン操作により、そのオフィスに含まれるルームへのエントリを表すドア31オブジェクトが表示される。このドア31への操作は、前記ドア31と同様である。ライブラリオブジェクトG 33はユーザが属するオフィスが管理するライブラリである。操作および内容は前述図6で説明した内容と同一である。クリップボードオブジェクトG 34はユーザが属するオフィスが管理するクリップボードである。操作および内容は前述図61で説明した内容と同一である。

【0046】＜デスクトップ管理システム用情報＞デスクトップ管理システムが動作する上で必要とする管理情報について説明する。

【0047】図14はデータ情報50のテーブルを表す図である。データ情報50はデスクトップ15が管理するアプリケーションプログラム26が作成するデータ18の形式、種類を登録し管理するための情報で、個々のデータ形式について次の項目が含まれる。51はデータ・フォーマットIDであり、個々のデータ形式毎に固有の番号である。52は拡張子であり、そのデータ形式のデータ18のファイルを作成する時に、ファイル名の最後につける文字である。53はアイコンファイル名であり、バインダ20内にデータ18を表示する際に使用するアイコンのデータが格納されているファイル名を表す。54はこのデータ形式のデータをライブラリデータとしても管理できる場合、そのライブラリデータのIDである後述するライブラリIDを表す。複数のライブラリIDを登録することが可能である。

【0048】図15はアプリケーション情報60のテーブルを表す図である。アプリケーション情報60は、デスクトップ15が管理するアプリケーション26についての情報であり、この情報に基づいて、後述のデスクトップ15のウィンドウ内にアプリケーション26のオブジェクトが表示される。登録されたアプリケーション26毎に次の情報が含まれる。61はアプリケーションIDであり、個々のアプリケーションプログラム毎に対して付けられた固有の番号である。62は実行モジュール名であり、アプリケーション26を実行する際に使用するプログラムの名を表す。63はアプリケーション名称

であり、アプリケーション26のオブジェクトの名称として表示される。64はアイコンファイル名であり、アプリケーション26をアイコン表示する際に使用するアイコンのデータが格納されているファイル名を表す。65は、そのアプリケーションプログラムが処理可能なデータ18のデータ形式を示すフォーマットID51と、処理コードと、その処理を表示するための名称との対応を示す処理リストである。66は処理リスト65のうち各フォーマットID51毎の処理リスト65の先頭のエントリで、これはアプリケーション26のデフォルトの処理を表しており、処理対象となるデータ18が指定されずにアプリケーション26が起動された場合の処理を示し、フォーマットID51とその処理リストからなる。67は処理リスト中の処理を行なう上で、処理対象となるデータ18が必要か否かを表し、“○”ならばデータが必要“×”ならば不必要、空白ならばどちらでも良いことを示す。

【0049】図16はライブラリ情報70のテーブルを表す図である。ライブラリ情報70は、デスクトップ管理システムが管理するライブラリデータ28についての情報であり、この情報に基づいて、後述のライブラリ17のウィンドウ内にライブラリバインダ27が表示される。登録されたライブラリバインダ27毎に次の情報が含まれる。71はライブラリIDであり、ライブラリバインダ27毎に対して付けられた固有の番号である。72はライブラリバインダ名であり、ライブラリデータ28が保存されている階層構造のファイルシステム中のライブラリバインダ27の所在（ディレクトリ）を示す。73は同一のデータをもったデータの集合であるライブラリの名称である。74は現在、ライブラリバインダ27下のライブラリデータ28に対してオープンされている回数を表す。

【0050】図17はユーザ情報80のテーブルを表す図である。ユーザ情報80は、1つのデスクトップ管理システムが稼動するシステムのユーザを管理する情報であり、各ユーザごとに次の情報が含まれる。81はユーザIDであり、各ユーザに対して付けられた固有の番号である。82はユーザの氏名などのユーザに関する任意の情報を表す。83はユーザのルーム13の所在（ディレクトリのパス）である。84はユーザのルーム13内のライブラリデータ28の更新モードを表す。このライブラリ更新モードについては後述する。

【0051】図18はオフィス情報90のテーブルを表す図である。オフィス情報90は、ネットワークなどで接続された他のデスクトップ管理システムが稼動しているオフィス14を管理する情報であり、各オフィス14毎に次の情報が含まれる。91はオフィスIDであり、個々のオフィスに14に対して付けられた固有の番号である。92はオフィス14の存在するホスト名である。93はオフィス名であり、ホスト内に存在するオフィス

14の名称を表す。94はドメインであり、オフィス14が属しているホストのドメインを表す。95はオフィス内のライブラリデータ28の更新モードを表す。ドメインとは、アーキテクチャが同じコンピュータの集合をいい、同一のドメインに属するコンピュータ上で稼動するデスクトップ管理システム間では、アプリケーションプログラムの作成したデータを特別な変換をする必要なく転送することができる。

【0052】図19はグループ情報100のテーブルを表す図である。グループ情報100は、データ18の参照、更新を制限するためのユーザの範囲を表すために複数のユーザをまとめた単位であるグループを管理する情報であり、各グループ毎に次の情報が含まれる。101はグループIDであり、各グループ毎に対して付けられた固有の番号である。102は各グループに属しているユーザのユーザIDである。

【0053】＜キャビネット管理用情報＞図20はオブジェクト管理情報110のテーブルを表す図である。オブジェクト管理情報110は、バインダ20内に含まれる個々のオブジェクトの情報を管理するためのファイルで、キャビネット16（ライブラリ17、ゴミ箱23も含む）とその下にあるバインダ20毎に存在する。また、オブジェクト管理情報110は、オフィス14のライブラリデータ28を管理するためにオフィス14毎に1つ存在する。データ管理情報110は1つのデータについて次の項目が含まれる。111はオブジェクトの種類（バインダ20、データ18、公開棚22、公開トレイ29、タグトレイ19、タグ21、ライブラリバインダ27、ライブラリデータ28）を示す。112はオブジェクトの内容であるファイル名またはディレクトリ名を表す。113はオブジェクトがデータ、またはライブラリデータである場合、そのデータのフォーマットIDを示す。また、オブジェクトがライブラリ17内のライブラリバインダである場合は、そのライブラリIDを示す。114はオブジェクトがデータである場合、そのデータを作成したアプリケーション26のアプリケーションIDであり、特にこれをクリエイタIDと呼称する。115はオブジェクトの漢字名称である。116はオブジェクトがデータである場合、そのデータのバイト単位のサイズであり、データが複数のファイル（フラグメント：後述）で構成されている場合は、その合計のサイズを表す。117は登録日付であり、そのオブジェクトが新規に作成された日付を表す。118は更新日付であり、そのオブジェクトが最後に変更された日付を表す。119は参照日付であり、そのオブジェクトが最後に参照された日付を表す。

【0054】120はオブジェクトの所有者のユーザIDを表す。121はオブジェクトの所有者のグループIDを表す。122はオブジェクトを最後に参照したユーザのユーザIDを表す。123はオブジェクトを最後に

更新したユーザのユーザIDを表す。124はアクセス・レベルであり、オブジェクトに対する参照、更新操作の許可、不許可の対象をデータの所有者ユーザID120、データの所有者が属するグループ121のユーザ、それ以外のユーザの3種類に分類して設定される。

【0055】125はオブジェクトがデータである場合、そのデータが作成／更新されたオフィス14のドメインを表す。126はオブジェクトが他のユーザに対して公開される公開トレイ29である場合に、その公開トレイの対象ユーザ、すなわち公開対象となるユーザのユーザIDを表す。127はオブジェクトが他のグループに対して公開される公開トレイ29である場合に、その公開トレイの対象グループ、すなわち公開対象となるグループのグループIDを表す。128はオブジェクトが公開オブジェクト30である場合に、公開先のユーザに対してそのオブジェクトの複写操作を制限する設定内容である公開を表す。129はオブジェクトが公開オブジェクト30である場合に、公開オブジェクト30の公開操作が有効な期限である公開期限を表す。130はオブジェクトがゴミ箱23内の置かれたデータである場合、データが一定機関経過すると自動的にゴミ箱23にて廃棄される期限であるデータ廃棄期限を表す。タグ参照先131はオブジェクトがタグ21である場合に、タグ21が指し示している操作オブジェクト（実体）の所在を示す。

【0056】図21は、オープンオブジェクト管理情報140のテーブルを表す図である。オープンオブジェクト管理情報140は、各ユーザのルーム13内の現在オープンされているバインダ20、データ18の状態を管理する情報であり、ルーム13毎に1つ存在する。

【0057】オープンオブジェクト管理情報140には1つのオブジェクトについて次の項目が含まれる。141はそのオブジェクトのファイル名である。142はそのオブジェクトをオープンしているユーザのユーザIDである。143はそのオブジェクトがアクセスレベル124の「参照」でオープンされているか、「更新」でオープンされているかを表すフラグである。144はそのオブジェクトがオープンされた日付を表す。

【0058】図22は、ライブラリ検索順番情報150を表す図である。ライブラリ検索順番情報150はライブラリデータ28を検索する際のライブラリ17の順番を表す情報であり、ルーム13毎に1つ存在する。

【0059】＜バインダのオープン＞図23はキャビネット16内のバインダ20をオープンする操作を表す図である。図23中、G37はポインティングデバイス5の画面上の位置を示すカーソル、G38は操作説明のためのポインティングデバイス5のボタン操作を表すマークであり、“★”でボタン1クリック“★”でボタン2クリック、“☆”でボタンが押されたことを“☆☆”でボタンが離されたことを表す。これらの記号は操作説明

を通して用いる。バインダ20のオープン操作は、キャビネットウィンドウG16中のバインダオブジェクトG17のアイコンヘカーソルG37を移動させてボタン2クリックすると、バインダ20の内容を表示するバインダウィンドウG39が現れる。このオープン操作は、操作オブジェクト全般に共通な操作である。

【0060】バインダ20のオープン処理を図24のフローチャートにより説明すると、S1で2クリックされたバインダオブジェクトG17の示すバインダ20のオブジェクト管理情報110を得て、S2で得られたオブジェクト管理情報110のフォーマットID113とデータ情報50からアイコンファイルを読み出して、S3で生成されたバインダウィンドウG39上にアイコン表示を行なう。S4でこれらの処理をオブジェクト管理情報110の全てのエントリに対して行い処理を終了する。この処理はキャビネット16、公開棚22、タグトレイ19、ライブラリバインダ27に共通である。デスクトップ15、出口25、ライブラリ17のオープン処理は後に述べる。

【0061】＜データのコピー＞図25はキャビネット16あるいはバインダ20間のデータ18のコピー操作を表す図である。データ18のコピー操作は、希望するコピー元のデータオブジェクトG18のアイコンヘカーソルG37を移動させて、ボタンを押したままカーソルG37を希望するコピー先のキャビネットウィンドウG16あるいはバインダウィンドウG39へ移動してボタンを離す操作（ドラグ／ドロップ操作）により完了する。

【0062】キャビネットにおけるデータ18のコピー処理方法を図26により説明すると、S10でコピー元のアイコンにボタンが押された段階でコピー元のデータ18が存在するバインダ20からデータ18のオブジェクト管理情報110を得ておく。S11で、得られたオブジェクト管理情報110からコピー元のデータ18が公開オブジェクト30で、かつ公開複写制限128の設定がある場合には、S12でコピー禁止の旨の表示を行い処理を中止する。S13でコピー先にボタンが離された後、S14でディスク容量などファイルのコピーが可能であるか検査し、コピー不可能であればS15でエラー表示を行ない処理を中止する。S16でコピー先のバインダ20のオブジェクト管理情報110に前もって得られたオブジェクト管理情報110のエントリを追加して、S17でデータ18の内容であるファイルをコピーすることにより処理を終了する。

＜アプリケーションの起動＞デスクトップ管理システムで管理されるアプリケーション26あるいはデータ18の示すオブジェクトをユーザが操作することによってアプリケーションが起動される。アプリケーションの起動は、アプリケーションの実行モジュール名62とそれに続いて処理されるデータ18のファイル名112とフォ



ーマットID51、処理リスト65中の処理方法を指示する文字列を付加することで実行される（例：／usr／local／bin／ezps／ezps-f 012-me）。

【0063】図3で、ルームウィンドウG01内のデスクトップオブジェクトG02を2クリックすると、図4のように、デスクトップウィンドウG14がオープンされて既に登録されているアプリケーション26をしめすアプリケーションオブジェクトG15が表示される。デスクトップ15のオープン処理は、アプリケーション情報60を得てアイコンファイルを読みだしウィンドウ上にアイコン表示を行なうことで終了する。

【0064】アプリケーションにより、データ18がオープンされると、データ18が存在するルームのオープンオブジェクト管理情報140にオープンされたデータ18のエントリが追加される。また、アプリケーションによってデータ18がクローズされると、前述オープンオブジェクト管理情報140のオープンされたデータ18のエントリが削除される。

【0065】次に、図を用いてユーザの操作別にデスクトップ管理システムにおけるアプリケーションの起動処理を説明する。

【0066】（1）アプリケーションオブジェクトのオープン操作による起動。

【0067】図27で、デスクトップウィンドウG14内のアプリケーションオブジェクトG15を2クリックする。

【0068】図28のフローチャートにより説明すると、S20で2クリックされたアプリケーションオブジェクトの示すアプリケーション26のアプリケーション情報60を得て、S21で処理リスト65の先頭エントリのフォーマットIDとそのデフォルト処理66の処理コードとを渡してアプリケーションプログラムを起動する。

【0069】（2）アプリケーションオブジェクトのオープン操作による起動。

【0070】図29で、デスクトップウィンドウG14内のアプリケーションオブジェクトG15をマウスのボタン1クリックによって選択すると、アプリケーションオブジェクトG15は選択されたことが判別できるように、アイコンの枠が太くなるなどの外観が変化する。この状態でメニューリストG41の中から「操作」ボタンG40をボタン1クリックによって選択すると、「操作」メニューG42が表示される。表示された「操作」メニューG42から、選択されたアプリケーションオブジェクトG15によって示されるアプリケーション26で実行したい処理を選択する。

【0071】図30のフローチャートにて説明すると、S30にて「操作」ボタンG40が選択されない場合にはS31で他の処理を行う。「操作」ボタンG40が選

択された場合には、S32でアプリケーションオブジェクトG15が選択された後にS33で選択されたアプリケーション26のアプリケーション情報60を得て、S34で得られたアプリケーション情報60の処理リスト65の先頭エントリにあるフォーマットIDがデフォルトとして選択されたことになり、前述フォーマットIDの処理リスト65の処理名称が「操作」メニューG42に表示される。「操作」メニューG42には、印刷機能などの既に作成されたデータを必要とする機能についてはアプリケーション情報60の処理データ必要67で検査されて表示されない。S35で「操作」メニューG42の1項目が選択されると、S36ですでにデフォルトとして選択されたフォーマットIDと選択された項目に対応する処理コードを渡して、すでに選択されたアプリケーションオブジェクトG15の示すアプリケーションプログラムを起動する。

【0072】（3）アプリケーションオブジェクトのドラッグ／ドロップ操作による起動。

【0073】図31で、デスクトップウィンドウG14内のアプリケーションオブジェクトG15をドラッグし、キャビネット（バインダ）ウィンドウG16内にドロップする。（1）の場合と同様に、ドラッグ／ドロップされたアプリケーションオブジェクトG15の示すアプリケーション26のアプリケーション情報60を得て、アプリケーション情報60の処理リスト65の先頭にあるフォーマットIDと、そのデフォルト処理66の処理コードを渡してアプリケーションプログラムを起動する。

【0074】（4）データオブジェクトのオープン操作による起動。

【0075】図32で、キャビネット（バインダ）ウィンドウG16内のデータオブジェクトG18を2クリックする。図33のフローチャートにて説明すると、S40で2クリックされたデータオブジェクトG18によって示されるデータ18のオブジェクト管理情報110からクリエイタID114とそのデータのフォーマットID113を得る。

【0076】次にS41で、前述得られたクリエイタID114と同じアプリケーションID61を持つアプリケーション情報60のエントリを得て、S42で前述得られたフォーマットID113と、前述得られたアプリケーション情報60のエントリにある前述得られたフォーマットIDのデフォルト処理66の処理コードと、データのファイル名112とを渡してアプリケーションプログラムを起動する。

（5）データオブジェクトのオープン操作による起動。

【0077】図34で、キャビネット（バインダ）ウィンドウG16内のデータオブジェクトG18をボタン1クリックによって選択すると、データオブジェクトG18は選択されたことが判別できるようにアイコンが反転表示されるなどの外観が変化する。この状態でメニュー

17

リストG 4 1の中から「操作」ボタンG 4 0をボタン1クリックによって選択すると、「操作」メニューG 4 2が表示される。表示された「操作」メニューG 4 2から選択したデータオブジェクトG 1 8に示されるデータ1 8に対して実行したい処理を選択する。

【0078】図35のフローチャートにて説明すると、ウィンドウ上の「操作」ボタンG 4 0が選択されない場合にはS 5 1で他の処理を行う。「操作」ボタンG 4 0が選択された場合にはS 5 2でデータオブジェクトG 1 8が選択され、オブジェクト管理情報1 1 0から、選択されたデータオブジェクトG 1 8の示すデータ1 8のフォーマットID 1 1 3とクリエイタID 1 1 4を得て、アプリケーション情報6 0からクリエイタID 1 1 4と同じアプリケーションID 6 1を待つエントリを得て、S 5 4で前述得られたアプリケーション情報6 0のエントリから、得られたフォーマットID 1 1 3に該当する処理リスト6 5の処理名称を「操作」メニューG 4 2として表示する。S 5 5で、「操作」メニューG 4 2の1項目が選択されると、S 5 6でデータ1 8のファイル名1 1 2と、既に得られたフォーマットID 1 1 3と、「操作」メニューG 4 2から選択された項目に対応する処理コードを渡して、前述得られたアプリケーション情報6 0の実行モジュール名6 2によってアプリケーションプログラムを起動する。

【0079】(6)データオブジェクトをアプリケーションオブジェクト上にドラッグ／ドロップして起動。

【0080】図36で、キャビネット（バインダ）ウィンドウG 1 6内の1つのデータオブジェクトG 1 8をドラッグし、デスクトップウィンドウG 1 4内のアプリケーションオブジェクトG 1 5にドロップする。

【0081】図37のフローチャートにて説明すると、データオブジェクトG 1 8がドラッグされると、S 6 0でオブジェクト管理情報1 1 0から、選択されたデータオブジェクトG 1 8によって示されるデータ1 8のファイル名1 1 2とフォーマットID 1 1 3を得る。S 6 1でドラッグされたデータオブジェクトG 1 8がデスクトップウィンドウG 1 4のアプリケーションオブジェクトG 1 5上に位置されると、S 6 2で、位置されたアプリケーション2 6のアプリケーション情報6 0を得て、S 6 3で、アプリケーション2 6がドラッグされているデータオブジェクトG 1 8によって示されるデータ1 8のフォーマットIDが処理可能なものであるならば、S 6 4で前述アプリケーションオブジェクトG 1 5のアイコン表示を反転させるなどの表示方法を変化させて、ユーザに対してアプリケーション2 6がドラッグされているデータオブジェクトG 1 8が示すデータ1 8を処理可能であることを知らせる。S 6 5でドラッグされたデータオブジェクトG 1 8がドロップされると、S 6 6で前述データ1 8が処理可能なアプリケーション2 6のアプリケーションオブジェクトG 1 5上にドロップされたかが検査され、

18

S 6 7で前述得られたアプリケーション情報6 0から、アプリケーションID 6 1に該当するフォーマットIDのデフォルト処理6 6の処理コードを得て、実行モジュール名6 2で示されるアプリケーション2 6に、フォーマットID 1 1 3と、処理コードと、データ1 8のファイル名1 1 2を渡してアプリケーションプログラムを起動する。

【0082】またこのデスクトップ管理システムでは、図15によって説明したように、1つのデータ形式のデータ1 8を複数のアプリケーション2 6によって処理することができる場合、また1つのアプリケーション2 6が複数のデータ形式のデータ1 8を処理できる場合に、データ1 8とそれを処理できるアプリケーションとの関係を管理する。このため、選択またはドラッグしたデータオブジェクトG 1 8を、どのアプリケーションオブジェクトG 1 5にドロップできるか、すなわちどのアプリケーション2 6がそのデータ1 8を処理可能であるかを操作を行なっているユーザに知らせることは有用なことがある。これを実現する手段として、次のガイダンス機能を提供する。

【0083】(6. 1)線による結合のガイダンス。

【0084】図38で、キャビネットウィンドウG 1 6あるいはバインドウィンドウG 3 9内の1つのデータオブジェクトG 1 8をグラフ（ポインティングカーソルを対象にあわせること）したときに、そのデータオブジェクトG 1 8とデスクトップウィンドウG 1 4内の処理可能なアプリケーションオブジェクトG 1 5を線によって結び付け、図39のごとく提示する。

【0085】この処理を図40のフローチャートにて説明すると、データオブジェクトG 1 8がグラフされると、そのデータオブジェクトG 1 8に対応するデータ1 8についてS 7 0でデータ管理情報1 1 0からファイル名1 1 2とフォーマットID 1 1 3を得る。S 7 1において、デスクトップ1 5内のアプリケーション2 6を検索する。S 7 2において、すでに検査した以外にアプリケーション2 6が存在しないと判断されたならば、処理可能なアプリケーションの提示を終了する。アプリケーションが存在した場合は、S 7 3においてそのアプリケーション2 6についてのアプリケーション情報6 0を得る。S 7 4において、そのアプリケーション情報6 0内に、グラフされているデータオブジェクトG 1 8の示すデータ1 8のフォーマットIDがあるか判定する。あるならば、グラフされたデータオブジェクトは、S 7 1で見いだされたアプリケーションで処理可能である。

【0086】データオブジェクトG 1 8を、デスクトップウィンドウG 1 4のアプリケーションオブジェクトG 4 3が処理可能であるならば、S 7 5で図39に示すようにデータオブジェクトG 1 8とアプリケーションオブジェクトG 4 3を線G 4 4で結び、ユーザに対してアプリケーションオブジェクトG 4 3の示すアプリケーショ

ン26が、グラフされているデータオブジェクトG18の示すデータ18を処理可能であることを知らせる。S71において、検索した1つのアプリケーション26についてS75の提示処理が終了すると、処理の制御をS71に戻し、S72でまだ検索していないアプリケーション26が存在しているならば、以降デスクトップ15内のアプリケーション26についてS73からS75の処理を繰り返す。これによって、S72の判断によって処理が終了すると、例えば図39で示すようにキャビネットウィンドウG16内でグラフされたデータオブジェクトG18と、それを処理することができるアプリケーション26のアプリケーションオブジェクトG43およびG45が線G44およびG46で結ばれる。

【0087】(6.2)ハイライト表示による結合のガイダンス。

【0088】図41で、キャビネットウィンドウG16あるいはバインドウィンドウG39内の1つのデータオブジェクトG18をグラフしたときに、そのデータオブジェクトG18を処理可能なデスクトップウィンドウG14内のアプリケーションオブジェクトG15をハイライトするなどの表示を変更して提示する。この処理を図41と、図42のフローチャートにて説明する。

【0089】データオブジェクトG18がグラフされると、そのデータオブジェクトG18に対応するデータ18についてS80でデータ管理情報110からファイル名112とフォーマットID113とを得る。S81においてデスクトップ15内のアプリケーション26を検索する。S82において、すでに検索した以外にアプリケーション26が存在しないと判断されたならば、処理可能なアプリケーションの提示を終了する。アプリケーションが存在した場合は、S83においてそのアプリケーション26についてのアプリケーション情報60を得る。S84において、そのアプリケーション情報60内に、グラフされているデータオブジェクトG18の示すデータ18のフォーマットIDがあるか判定する。あるならば、グラフされたデータオブジェクトは、S81で見いだされたアプリケーションで処理可能である。たとえば、データオブジェクトG18を、デスクトップウィンドウG14のアプリケーションオブジェクトG47が処理可能であるならば、S85で図41に示すようにアプリケーションオブジェクトG47の表示をハイライト表示し、ユーザに対してアプリケーションオブジェクトG47の示すアプリケーション26が、ドラグされているデータオブジェクトG18の示すデータ18を処理可能であることを知らせる。S81において検索した1つのアプリケーション26についてS85の提示処理が終了すると、処理の制御をS81に戻し、S82でまだ検索していないアプリケーション26が存在しているならば、以降デスクトップ15内のアプリケーション26についてS83からS85の処理を繰り返す。これによ

て、S82の判断によって処理が終了すると、例えば図41で示すようにキャビネットウィンドウG16内でグラフされたデータオブジェクトG18を処理することができるアプリケーション26のアプリケーションオブジェクトG47およびG48がハイライト表示される。

【0090】(6.3)引力によるガイダンス。

【0091】図43で、キャビネットウィンドウG16あるいはバインドウィンドウG39内の1つのデータオブジェクトG18をドラグして移動させたとき、そのドラグしているデータオブジェクトG18がデスクトップウィンドウG14内の処理可能なアプリケーションオブジェクトG15からの定められた距離に達した時点で、デスクトップ管理システムによって、ドラグされたデータオブジェクトG18の表示位置を自動的に補正して提示する。この処理を図43と図44のフローチャートにて説明する。

【0092】データオブジェクトG18がグラフされると、そのデータオブジェクトG18に対応するデータ18についてS90でデータ管理情報110からファイル名112とフォーマットID113とを得る。S91において、ドラグされたデータオブジェクトG49とアプリケーションオブジェクトG50との重なり面積が、ドラグされたデータオブジェクトG49の表示面積のある一定の割合に達したかを判断する。S91において、ドラグされたデータオブジェクトG49とアプリケーションオブジェクトG50のアイコンとの重なり面積が、ドラグされたデータオブジェクトG49の表示面積のある一定の割合に達していないと判断した場合は制御をS91に戻す。

【0093】S91において、ドラグされたデータオブジェクトG49とアプリケーションオブジェクトG50のアイコンとの重なり面積が、ドラグされたデータオブジェクトG49の表示面積のある一定の割合に達したと判断されたならば、S92においてアプリケーションオブジェクトG50の示すアプリケーション26のアプリケーション情報60を得る。

【0094】S94において、アプリケーションオブジェクトG50の示すアプリケーション26のアプリケーション情報60内に、ドラグされているデータオブジェクトG49の示すデータ18のフォーマットIDがあるかテストする。あれば重なり合ったアプリケーションによりデータが処理可能であり、可能と判断されたならば、S95において、ドラグされたデータオブジェクトG49の表示位置を補正し、G51のようにデータオブジェクトG49とアプリケーションオブジェクトG50が完全に重なるようにし、同時にS96でアプリケーションオブジェクトG50をハイライト表示する。この処理によってユーザは、ドラグしているデータオブジェクトG49があたかも引力によってアプリケーションオブジェクトG50に引き寄せられたように感じ、そのアプ

リケーションオブジェクトG50の示すアプリケーション26がドラッグしているデータオブジェクトG49の示すデータ18を処理可能であると判断することができる。

【0095】S96に続いて、S97において、ドラッグされたデータオブジェクトG49がアプリケーションオブジェクトG50上にドロップされたかを判断する。S97において、ドロップされないと判断された時は、制御をS91に戻す。S97において、ドロップされたと判断された時は、S98において、S92で得られたアプリケーション情報60から、アプリケーションID61に該当するフォーマットIDのデフォルト処理66の処理コードを得て、アプリケーションオブジェクトの示すアプリケーション26にフォーマットIDと処理コードとデータオブジェクトG18が示すデータ18のファイル名112を渡してアプリケーション26を起動する。

【0096】(6.4)反力によるガイダンス。

【0097】図45で、キャビネットウィンドウG16あるいはバインダウィンドウG39内の1つのデータオブジェクトG18をドラッグして移動させたとき、そのドラッグしているデータオブジェクトG18がデスクトップウィンドウG14内の処理不可能なアプリケーションオブジェクトG15からの定められた距離に達した時点で、デスクトップ管理システムによって、ドラッグしたデータオブジェクトG18の表示位置を自動的に補正して提示する。この処理を図45と、図88のフローチャートにて説明する。

【0098】データオブジェクトG18がグラブされると、そのデータオブジェクトG18に対応するデータ18についてS90でデータ管理情報110からファイル名112とフォーマットID51を得る。S91において、ドラッグされたデータオブジェクトG53とアプリケーションオブジェクトG52との重なり面積が、ドラッグされたデータオブジェクトG53の表示面積のある一定の割合に達したかを判断する。

【0099】S91において、ドラッグされたデータオブジェクトG53とアプリケーションオブジェクトG52のアイコンとの重なり面積が、ドラッグされたデータオブジェクトG52の表示面積のある一定の割合に達していないと判断した場合は制御をS91に戻す。S91において、ドラッグされたデータオブジェクトG53とアプリケーションオブジェクトG52のアイコンとの重なり面積が、ドラッグされたデータオブジェクトG52の表示面積のある一定の割合に達したと判断されたならば、S92においてアプリケーションオブジェクトG52の示すアプリケーション26のアプリケーション情報60を得る。

【0100】S94において、アプリケーションオブジェクトG52の示すアプリケーション26のアプリケー

ション情報60に、ドラッグされているデータオブジェクトG53の示すデータ18のフォーマットIDがあり、データ18を処理可能であると判断されたならば、以下(6.3)と同様のアプリケーション26の起動に制御を移す。反対に、S94において、アプリケーションオブジェクトG52の示すアプリケーション26が、ドラッグされているデータオブジェクトG53の示すデータ18を処理不可能であると判断されたならば、S93において、ドラッグされたデータオブジェクトG53の表示位置を補正し、G54のようにデータオブジェクトG53とアプリケーションオブジェクトG52の重なりを減少あるいはなくす。この処理によってユーザは、ドラッグしているデータオブジェクトG53があたかも反力によってアプリケーションオブジェクトG52から引き離されたように感じ、そのアプリケーションオブジェクトG52の示すアプリケーション26がドラッグしているデータオブジェクトG53の示すデータ18を処理不可能であると判断することができる。

【0101】(7)複数のデータオブジェクトをアプリケーションオブジェクト上にドラッグ/ドロップして起動。

【0102】図38で、キャビネット(バインダ)ウィンドウG16内の複数のデータオブジェクトG18をドラッグし、デスクトップウィンドウG14内のアプリケーションオブジェクトG15にドロップする。(6)と同様に、ドラッグされたそれぞれのデータオブジェクトG18のフォーマットID51を得る。ドラッグされたデータオブジェクトG18がデスクトップウィンドウG14のアプリケーションオブジェクトG15上に位置されると、アプリケーション情報60を検索して、アプリケーションオブジェクトG15の示すアプリケーション26に、ドラッグされている全てのデータオブジェクトG18のフォーマットIDがあるならば処理可能であるとして、アプリケーションオブジェクトG15のアイコン表示をハイライトさせるなどの表示方法を変化させてユーザに対してそのアプリケーション26がドラッグされている全てのデータオブジェクトの示すデータ18を処理可能であることを知らせる。

【0103】前にハイライト表示されたアプリケーションオブジェクトG15上にドラッグされた複数のデータオブジェクトG18がドロップされると、アプリケーションオブジェクトG15が示すアプリケーション26のアプリケーションID61とデータオブジェクトの示すデータ18のフォーマットID51とからアプリケーション情報60を検索し、アプリケーションID61に該当する実行モジュール名62とフォーマットID51のデフォルト処理66の処理コードを得て、フォーマットID51と処理コードと複数のデータ18のファイル名112を渡してアプリケーション26を起動する。

【0104】<公開操作>通常、ユーザのルーム13内

のキャビネット16, バインダ20, データ18を他のユーザが参照, 更新することは許されていない。ユーザが自分の所有するキャビネット16, バインダ20, データ18を他のユーザに対して参照あるいは更新することを許可する操作を「公開操作」と呼ぶ。また、「公開操作」を行なう或は行なわれたキャビネット16, バインダ20, データ18を公開オブジェクト30と呼ぶ。次に、「公開操作」を行なうための操作および処理を説明する。

【0105】まず、公開対象とするユーザあるいはグループ毎に公開トレイ29を公開棚22上に作成する。作成方法は、図46において公開棚ウィンドウG26上でウィンドウ上の新規作成ボタンG55を選択し、問い合わせの入力ウィンドウG56に公開する相手のユーザの名称またはユーザID81、あるいはグループの名称またはグループID101を入力すると、公開棚ウィンドウG26内に新規の公開トレイ29を示す公開トレイオブジェクトG27が作成され表示される。

【0106】続いて、公開対象のユーザあるいはグループの公開トレイオブジェクトG27をオープンし、公開トレイウィンドウG28を表示する。ここでバインダ20間のデータ18のコピー操作と同様に、公開したいデータオブジェクトG18またはバインダオブジェクトG17をドラッグして公開トレイウィンドウG28上へドロップする。この操作によりドロップされた公開オブジェクトG29が公開トレイウィンドウG28上に表示されて公開操作が行なわれたことを表す。また、公開したオブジェクトに対して他のユーザの複写操作を制限する設定や、公開を許可する期限を設定することが可能である。

【0107】設定操作は、図47において公開トレイウィンドウG28の操作ボタンG57を押してから設定したい公開オブジェクトG29を1クリックすることで、問い合わせの入力ウィンドウG58が表示されるので設定項目を入力することで設定が完了する。既に設定がされている場合には入力ウィンドウG58表示時に設定項目欄内に設定内容が表示される。デスクトップ管理システムでの公開トレイ29作成処理を図48のフローチャートと図46にて説明する。

【0108】公開棚ウィンドウG26上の新規作成ボタンG55が選択されたらS100で公開対象を入力する入力ウィンドウG56を作成し、S101にてユーザからの入力待ち、S102にてユーザ名称で入力された場合にはS103でユーザ情報80を検索してユーザID81を得る。S104にてユーザID81で入力された場合には、S105でユーザ情報80を検索して入力されたユーザID81が存在するか検査する。S106にてグループ名称で入力された場合には、S107でグループ情報100を検索してグループID101を得る。S108にてグループID101で入力された場合

には、S109でグループ情報100を検索して入力されたグループID101が存在するか検査する。

【0109】S110でユーザID81あるいはグループID101が存在する場合、S111で公開棚22のオブジェクト管理情報110に新規の公開トレイ29のエントリを作成して、公開先ユーザID126あるいは公開先グループID127に前に得られたユーザID81あるいはグループID101を代入する。S112にて作成された公開トレイ29をしめす公開トレイオブジェクトG27を公開棚ウィンドウG26内に表示して処理を終了する。ユーザID81あるいはグループID101が得られなかった場合には処理を中止する。

【0110】次に、デスクトップ管理システムでの「公開操作」処理を図49のフローチャートと図47にて説明する。

【0111】公開したいオブジェクトがドラッグされた時点で、S120にてドラッグされたオブジェクトのオブジェクト管理情報110を得る。S121でドラッグされたオブジェクトが公開トレイウィンドウG28上にドロップされた場合には、S122で、公開トレイウィンドウG28の示す公開トレイ29のオブジェクト管理情報110に、新規の公開オブジェクト30のエントリを作成して前に得られているオブジェクト管理情報110を代入する。S123で、ドラッグされたオブジェクトを公開オブジェクトG29として公開トレイウィンドウG28上に表示して処理を終了する。

【0112】次に、デスクトップ管理システムでの公開オブジェクト30に対する公開複写制限設定および公開期限設定処理を図47と図50のフローチャートにて説明する。

【0113】まず、S130で公開トレイウィンドウG28上の操作ボタンG57が選択されたら、S132で公開オブジェクトG29の選択入力待ち、S133で選択された公開オブジェクトG29のオブジェクト管理情報110を得る。

【0114】次に、S134で問い合わせの入力ウィンドウG28を作成し、入力欄上に前に得られたオブジェクト管理情報110から公開複写制限128、公開期限129の内容を表示してS135で入力待ち。設定入力後、S136で入力された公開複写制限128、公開期限129の内容を前に得られたオブジェクト管理情報110に代入し、設定の完了したオブジェクト管理情報110を公開トレイ29内のオブジェクト管理情報110へ代入して処理を終了する。

【0115】＜他のユーザのデータへのアクセス＞ユーザが他のユーザのデータをアクセスする場合の操作および処理を図2で説明する。

【0116】まず、自分のルーム13内の出口25をオープンすると、自分のオフィス14内の他のユーザのルーム13を表すドア31と、他のオフィス14と、自分

のオフィス14が管理するライブラリ17が表示される。他のオフィス14をオープンすれば、オープンされたオフィス14のユーザのドア31とライブラリ17が再度表示される。希望するユーザのドア31をオープンすると、オープンされたルーム13のユーザが自分に対して公開操作を行なっている公開オブジェクト30が表示される。この公開オブジェクト30に対する操作は自分のルーム13内の操作オブジェクトと同様の操作が可能である。ただし、公開オブジェクト30に対して公開複製制限の設定がされている場合には、その公開オブジェクト30の内容を複製する操作を行なうことが禁止される。また、公開オブジェクト30に対して公開期限の設定がされている場合には、公開期限を過ぎた公開オブジェクト30はウィンドウ上に表示されない。

【0117】デスクトップ管理システムでの出口25オープン処理を図3、図11、図12、図13さらに図51のフローチャートにて説明する。

【0118】出口オブジェクトG09がオープンされると出口ウィンドウG30が表示され、S140でユーザ情報80およびオフィス情報90から他のユーザおよびオフィス14の名称を得る。S141で出口ウィンドウG30上にドア31を示すドアオブジェクトG31、オフィス14を示すオフィスオブジェクトG32およびライブラリ17を示すライブラリオブジェクトG33の表示を行ない、S142で入力待。

【0119】S143でライブラリオブジェクトG33に対してオープン操作が行われた場合には、S144で後述するライブラリ17のオープン処理を行い処理を終了する。S145でオフィスオブジェクトG32に対してオープン操作が行われた場合には、S146でオフィス情報90からホスト名92を得てオープンされたホストに対してユーザ情報80を要求することによりユーザ情報80を得てS141へ戻り、図12のオフィスウィンドウG35にそのオフィスに含まれるユーザのルームへのエントリを示すドア31のドアオブジェクト30の表示を行ない入力待。

【0120】S147でドアオブジェクトG31に対してオープン操作が行われた場合には、S148でオープンされたドア31のユーザ公開棚22のオブジェクト管理情報110を得て、S149で操作しているユーザ用の公開トレイ29を検索する。操作しているユーザ用の公開トレイ29が存在する場合には、S150で前記公開トレイ29のオブジェクト管理情報110を得て、S151で公開期限が過ぎていない公開オブジェクト30だけを図13のルームウィンドウG44に表示する。S149で公開トレイ29が存在しない場合には、S125でその旨を表示して処理を終了する。

【0121】＜タグ＞タグ21は、階層の深いバインダ20あるいはデータ18を操作する際に、その途中の階層バインダ20のオープン操作を複数回行なうことなく

目的のバインダ20あるいはデータ18を操作することを可能とするオブジェクトである。

【0122】次に、タグ21の生成方法および処理方法について説明する。

【0123】図3のタグトレイ19を示すタグトレイオブジェクトG05をオープンすると、図8のようにタグオブジェクトG25が表示されたタグトレイウィンドウG24が表示される。表示されたタグオブジェクトG25の実態は「キャビネット」下に存在するバインダ20およびデータ18であり、表示されたタグオブジェクトG25の示すタグ21に対する操作は実態であるバインダ20およびデータ18に対する操作と同じ処理が行なわれる。ただし、削除操作に関してはタグ21自身が削除されるのであり、実態であるバインダ20あるいはデータ18は削除されない。タグ21の新規作成方法は、バインダ20間のデータ18コピー操作同様に作成したいタグ21の実態であるバインダ20のバインダオブジェクトG17あるいはデータ18のデータオブジェクトG18をドラッグしてタグトレイウィンドウG24へドロップすることで完了する。この操作によりタグトレイウィンドウG24上にドラッグされたバインダオブジェクトG17あるいはデータオブジェクトG18のアイコンデータを表示したタグオブジェクトG25が表示されてタグ21が新規作成されたことを表す。タグ21の新規作成処理を図52のフローチャートにて説明する。

【0124】S160でバインダオブジェクトG17あるいはデータオブジェクトG18がドラッグされた時点で、バインダオブジェクトG17の示すバインダ20あるいはデータオブジェクトG18の示すデータ18が存在するバインダ20のオブジェクト管理情報110を得る。S161でタグトレイウィンドウG24上にボタンが離された時に、S162でそのタグトレイウィンドウG24が示すタグトレイ19のオブジェクト管理情報110に前に得られたオブジェクト管理情報110のエントリを追加する。さらにS163でオブジェクト管理情報110内のタグ参照バインダ名131にバインダ20あるいはデータ18が存在するバインダ20の名称を代入し、S164で作成されたタグオブジェクトG25をタグトレイウィンドウG24上に表示して処理を終了する。

【0125】次に、タグオブジェクトG25のオープン操作処理を図53のフローチャートにて説明する。

【0126】タグオブジェクトG25に対して操作が行なわれた場合に、S170でタグトレイ19内に存在する操作されたタグオブジェクトG25が示すタグ21のオブジェクト管理情報110を得る。S171で実態であるバインダ20あるいはデータ18が存在するバインダ名131、およびバインダ20あるいはデータ18のオブジェクト管理情報86を得る。

【0127】S172で前記オブジェクト管理情報が得

られた場合には、その後の処理はS173でバインダ20やデータ18に対する処理操作と同一の処理が行われる。S172で前記オブジェクト管理情報が得られなかった場合には、S174でタグが示すデータあるいはバインダが存在しない旨の表示を行い処理を終了する。

＜データの廃棄＞作成されたバインダ20、データ18を廃棄するには、バインダ20間のデータ18のコピー操作と同様に、図3のルームウィンドウG01に表示されたゴミ箱オブジェクトG07のオープン操作によって表示されるゴミ箱ウィンドウG59内に廃棄したいバインダ20のバインダオブジェクトG17、データ18のデータオブジェクトG18をドロップすることでバインダオブジェクトG17、データオブジェクトG18がゴミ箱ウィンドウへ移動する。デスクトップ管理システムにはデータ廃棄を監視するプロセスが常時動作しており、移動されたバインダオブジェクトG17とそれが示すバインダ20と、データオブジェクトG18とそれが示すデータ18は一定期間保存の後に消去される。

【0128】データの廃棄処理を図5、図54、図55のフローチャートにて説明する。

【0129】S180でデータ18のデータオブジェクトG18がグラブされた時点で、そのデータオブジェクトG18が存在するバインダ20のオブジェクト管理情報110を得る。S181でゴミ箱ウィンドウG59上にボタンが離れた時に、S182で前に得られたオブジェクト管理情報110に更新不許可の設定がある場合には、S183にて廃棄不可の表示を行い処理を終了する。

【0130】上述以外の場合であれば、廃棄しようとするデータが他でオープンされていないかS184でテストし、オープンされているなら廃棄不可であってS183に進み、オープンされていないならS185にて前述データ18をゴミ箱23へコピー処理を行う。S186で正常にコピー処理が終了した場合には、S187で前記データ18が存在するバインダ20のオブジェクト管理情報110の前記データ18のエントリを削除して前記データ18のファイルも消去する。

【0131】最後に、S188でゴミ箱23のオブジェクト管理情報110に追加された前記データ18のエントリのデータ廃棄期限に、現在の時間から一定期間後の日付を代入して処理を終了する。

【0132】データ廃棄監視プロセスの処理方法を図56のフローチャートにて説明する。まず、S190でゴミ箱23のオブジェクト管理情報110を得る。S191で、各エントリに対してデータ廃棄期限の過ぎているオブジェクト管理情報110のエントリを検索する。存在した場合にはS192で前述エントリと前述エントリ内のファイル名が表すファイルを削除することによって、削除された前述エントリの示すデータ18またはバインダ20が削除され、そのデータオブジェクトG18

またはバインダオブジェクトG17はゴミ箱ウィンドウG59の表示されない。

【0133】S193でこれらの処理を前述オブジェクト管理情報110の全てのエントリに行ったかテストし、すべて終了した後は、S194およびS195にて一定期間経過するのを待ち、S190へ戻って同じ処理を繰り返す。

【0134】＜ライブラリ管理＞ライブラリに管理されるデータとは、アプリケーションプログラムがデータを処理するために他のアプリケーションと共通的に参照するための補助データをいう。例えば文書データを処理するアプリケーションプログラムを例にすると、文書データには、文字データとその文字データによって文書を整形するためのコマンドが含まれ、ライブラリデータとしてそのコマンドを解釈して文書を整形するための情報を持った書式データ、あるいはその文字データを表示するためのフォントデータを独立したデータとして管理する。これによって1つの文書データをアプリケーションプログラムが処理し、CRT上に表示したり印刷出力する場合に、例えば補助データとして使用するライブラリデータとしてのフォントデータや、書式データを切替えることによって文書データに変更を加えることなくその体裁を容易に変更することができる。また、1つの書式データを複数の文書データで共通に使用するような場合も、それぞれの文書データ中に書式データを含める必要がなくなるため文書データの肥大を防止することができる。これらのライブラリに管理される状態のデータをライブラリデータ28という概念でとらえる。

【0135】データ18には必要なライブラリデータ28を特定する情報が付加されており、アプリケーション26とその使用するデータ18で必要なライブラリデータ28が決められ要求される。アプリケーション26が新規にデータ18を作成する場合にも、既存のライブラリデータ28を利用できる。

【0136】次に、図に従ってライブラリの管理を説明する。

【0137】図7のように、ライブラリバインダウィンドウG22内に表示されるライブラリデータオブジェクトG23が示すライブラリデータ28は、キャビネット16下で管理するデータ18とオブジェクト管理情報110自体の形式は同様である。これらは通常のデータ18としてキャビネット16下で管理することもできる。デスクトップ管理システムは、キャビネット16下で管理されるデータ18をライブラリデータ28としても管理する場合、ライブラリデータ28としての管理はその（ユーザがつけた）名前のみを保持し、ライブラリデータ28が必要になった時点でその名前によってその実体を名前によってキャビネット16下のデータ18を参照するようにしても良い。

【0138】図3、図11のように、デスクトップ管理

システムが管理するライブラリ17は、そのユーザのルームウィンドウG01内に表示されるライブラリと、または出口ウィンドウG30内に表示されるライブラリとがある。出口ウィンドウG30に表示されるライブラリオブジェクトG33によって表されるライブラリは、ルーム13の所属するオフィス14に付属するライブラリであり、ユーザのルームウィンドウG01内に表示されるライブラリオブジェクトG04によって表されるライブラリは、そのルーム13に付属するライブラリである。このようにライブラリ17はその管理の形態によって次の2つに分類される。

【0139】(1) ルーム13に付属するライブラリ17。

【0140】ルーム13のオブジェクトとして登録される個人用のライブラリ17で、その有効範囲は一利用者となる。また、前述のデータ18の公開操作と同様の処理を施すことにより、他のユーザからもアクセスできるようにすることで、ライブラリ17の有効範囲を任意に指定された複数利用者の集合とすることもできる。

【0141】(2) オフィス14に付属するライブラリ17。

【0142】オフィス14のオブジェクトとして登録されるライブラリ17で、そのオフィス17に含まれるルーム13のユーザのみが共通にアクセスできるライブラリ17であり、その有効範囲はシステムによって恒久的に指定された複数利用者の集合となる。また、前述の公開操作により、ライブラリ17の有効範囲を全利用者にすることもできる。

【0143】ライブラリ17は、図16に示すライブラリ情報70(ライブラリデータ28の種類)毎にライブラリバインダ27を管理しており、ライブラリ17をオープンすると、図6のようにライブラリウィンドウG19が表示される。1つのライブラリバインダ27には、フォーマットID51が同じライブラリデータ28のみが格納されており、ライブラリバインダ27が属性として待つライブラリID71は、格納されているライブラリデータ30のデータ情報50のライブラリID54と同一である。

【0144】<ライブラリバインダの作成>ユーザがルーム13内のライブラリ17に新たなライブラリバインダ27を作成する手順および処理を図57と図58のフローチャートを用いて説明する。

【0145】ユーザがルーム13内のライブラリ17に新たなライブラリバインダ27を作成する場合、S200でライブラリウィンドウG19のライブラリバインダの生成ボタンG60を選択する。デスクトップ管理システムは、S202でデータ情報からライブラリID54の登録があるエントリだけを得て、S203ではライブラリ選択ウィンドウ101にて、前述データ情報50のみの一覧を表示したライブラリ選択ウィンドウG61を

表示する。S204でユーザがそのうちの1つを選択することで、S205でライブラリ17内のライブラリ情報70のエントリを追加作成して前述選択されたデータ情報50に新規作成されたライブラリID71を代入して、S206でライブラリ17のウィンドウ上にライブラリバインダ27のライブラリバインダオブジェクトG20を追加表示する。

【0146】アプリケーション26を新規に登録する際に、新たなライブラリが必要な場合、必要なグループ(ライブラリデータ31種別)のライブラリバインダ27、およびライブラリデータ28が登録されるようにすることもできる。この時、アプリケーション26の登録において、そのライブラリ17のグループとともにデスクトップ管理システムにライブラリ情報70が登録され、この時点でアプリケーション26が提供するデフォルトのデータ18がオフィス14に付属するライブラリ17として登録される。ライブラリ17のグループが登録されると、そのグループ毎にライブラリバインダ27が生成されそのグループのデータを格納、すなわちライブラリデータ28として登録、管理される。

【0147】<データのライブラリへの登録>キャビネット16下のデータ18をライブラリデータ28として登録する操作および手順を図59のフローチャートを用いて説明する。

【0148】キャビネット16下のデータ18をライブラリデータ28として登録する場合、図25に示す通常のデータ18の複写(移動)と同様の操作になる。ユーザは、登録先のライブラリ17のライブラリバインダ27を示すライブラリバインダオブジェクトG20をオープンし、ライブラリバインダウィンドウG22を表示する。ライブラリに登録するバインダ20内のデータ18を表すデータオブジェクトG18をドラッグし、ライブラリバインダウィンドウG22内にドロップする。

【0149】デスクトップ管理システムは、S210でコピー元のバインダ20のオブジェクト管理情報110を得て、次にS211で、前に得られたオブジェクト管理情報110のフォーマットID113と同じフォーマットID51を持つデータ情報50のエントリを得る。さらにS212で、ライブラリ17のライブラリ情報70からデータオブジェクトG18がドロップされたライブラリバインダウィンドウG22の示すライブラリバインダ27のライブラリID71を得る。S213で、S211で得たデータ情報50のライブラリID54とライブラリID71とを比較し、異なっている場合はS214で登録不可能な操作の表示をし、処理を中止する。

【0150】S214でライブラリID54とライブラリID71が一致している場合は、S215でそのデータ18をコピー元のバインダ20からライブラリバインダ27に既に説明したコピー処理を行い、S216でライブラリバインダウィンドウG22内にそのデータ18



をライブラリデータオブジェクトG23として表示する。

【0151】<ライブラリの検索>アプリケーション26が、デスクトップ管理システムが管理するライブラリデータ28を使用する場合、すでに公知の任意のプログラム間通信手段を使用して、デスクトップ管理システムにメッセージを送る。このメッセージには「ライブラリデータの要求」というコマンドと、要求するライブラリID71、ライブラリデータ28を特定できる場合は特定する情報、例えばライブラリデータ28の名称と、さらにライブラリデータ28を使用するデータ18を特定する情報が含まれている。

【0152】ライブラリデータ28の検索処理を図60のフローチャートで説明する。

【0153】アプリケーション26からのメッセージを受けたデスクトップ管理システムは、S220で、メッセージを解析してライブラリデータ28を使用するデータ18を特定する情報から、そのデータが属するルーム13のライブラリ情報70を得て、S221で前に得られたライブラリ情報70から要求されているライブラリID71のライブラリデータ28を管理するライブラリバインダのオブジェクト管理情報110を得る。

【0154】S222でライブラリデータ28を特定する情報が設定されているかテストし、設定されている場合、S223で、得られたライブラリバインダ27のオブジェクト管理情報110からライブラリデータ28を検索する。S226で、得られたライブラリデータ28を前記通信手段を使用してアプリケーション26に通知する。

【0155】一方、S222で、ライブラリデータ28を特定する情報が設定されていないと判定された場合は、アプリケーション26が、ユーザの選択する新たなライブラリデータ28を要求する場合である。この場合、デスクトップ管理システムは、S224で、得られたライブラリバインダ27のオブジェクト管理情報110からの前記ライブラリバインダ27のウィンドウをオープンして、前述ライブラリバインダ27内のライブラリデータオブジェクトG23を表示する。S225でユーザによるライブラリデータ28への選択入力を待つ。S226で選択されたライブラリデータ28を前記通信手段を使用してアプリケーション26に通知する。

【0156】<データ要求によるライブラリの検索>データ18を処理中のアプリケーション26がライブラリデータ28を要求する場合、アプリケーション26の処理対象となっているデータ18が存在するルーム13のライブラリ17から検索され、そこに存在しない場合にはそのルームの属するオフィス14のライブラリ17から検索される。すなわち、他のオフィス14のデータ18を処理中のアプリケーション26よりライブラリデータ28を要求すると、データ18が存在するオフィス1

4のなかで、データ18の所有者のルーム13のライブラリ17、前述所有者のルーム13が属するオフィス14のライブラリ17という順で検索される。他のユーザのルーム13から自分のルーム13にコピーしてきたデータ18をアプリケーション26がオープンした場合は、オープンしたデータ18についてのライブラリデータ28要求には、自分のルーム13のライブラリ17、そのルームの含まれるオフィス14のライブラリ17の順で検索される。他のルーム13から自分のルーム13にデータをコピーする際に参照するライブラリデータ31を自分のルーム13のライブラリ17に自動的に複写するようにすることも可能である。この場合、複写されるライブラリデータ28と同じライブラリID71を持つライブラリバインダ27内に複写される。また、データ18のライブラリデータ28の検索順番は図22に示すライブラリ検索順番情報150によりオフィスID91あるいはユーザID81を得ることで特定のルーム13のライブラリ17またはオフィス14のライブラリ17を指定することもできる。ライブラリ検索順番情報150はルーム13毎の情報であり、検索順番はユーザが任意に設定することができる。

【0157】図22の例では、2つまで設定するようにしているが、任意の数の検索順番を設定可能である。ライブラリ検索順番情報150が設定されていない場合は、前述の順番で検索される。

【0158】<ライブラリデータの更新>ライブラリデータ28はアプリケーション26によって作成、編集、削除等の更新を通常のデータ18と全く同様に行なうことができる。

【0159】図6に示すように、ライブラリ情報70にはライブラリバインダ27内のライブラリデータ28がいくつオープンされているかを示すオープン回数74の情報があり、また、図17、図18に示すように、ユーザ情報80、オフィス情報90にはそれぞれルーム13、オフィス14のライブラリ17が更新モードでオープンされているかを示すライブラリ更新モード84およびライブラリ更新モード95（以下、まとめて「ライブラリ更新モード」と記す）の情報があり、ライブラリデータ28の更新を管理することができる。

【0160】ルーム13またはオフィス14で最初に更新モードでオープンしたライブラリ17に限り更新モードでオープンが可能であり、前述オープンされた時点で「ライブラリ更新モード」が設定されてライブラリデータ28の更新が可能となる。

【0161】アプリケーション26がデータ18を要求する場合、データ18が存在するルーム13のライブラリ17に「ライブラリ更新モード」が設定されている状態でもデータ18のオープンが可能であるが、アプリケーション26からのライブラリデータ28の要求に対する応答には「参照不可」が返る。つまり、ライブラリデ

ータ28のみ欠如したデータ18がアプリケーション26にわたされる。

【0162】「ライブラリ更新モード」を設定したアプリケーション26がライブラリデータ28をクローズした時点で、「ライブラリ更新モード」の設定が解除され、そのライブラリ17の参照が可能になる。参照モードでのみオープンされているライブラリバインダ27のライブラリデータ28に対しても同時に複数オープンされているライブラリバインダ27のライブラリデータ28を更新操作を禁止することもできる。いくつかのライブラリデータ28がオープンされていれば更新操作禁止とするオープン回数74の設定は任意であり、オープン回数74により判断することができる。

【0163】図6のG21に示すように、デスクトップ管理システムはライブラリバインダ27のオープン回数74を表示する機能を有し、ユーザはオープン回数74を知ることができる。

【0164】＜クリップボード＞アプリケーション26はデータ18またはその一部を、そのアプリケーション内、または他のひとつまたは複数のアプリケーションに複写または移動させることがあり、そのために作成するデータ、およびそのデータを一時的に格納する場所をそれぞれクリップデータ、クリップボードという概念で管理する。このクリップボード24は、格納したクリップデータ32を表示するためのウィンドウを持つ。クリップデータ32は、キャビネット16やバインダ20内のオブジェクトと同様に、図20で示したオブジェクト管理情報によって管理される。クリップデータ32には、そのクリップデータ32を作成したアプリケーション26によって、そのデータ種別を表すフォーマットID113が設定される。

【0165】クリップデータ32を要求するアプリケーション26は、デスクトップ管理システムにフォーマットID113によって要求するデータの種別を特定する。次に、図にしたがって説明する。

【0166】図3、図11のように、デスクトップ管理システムが管理するクリップボード24は、そのユーザのルームウィンドウG01内に表示されるクリップボードと、出口ウィンドウG30内に表示されるクリップボードとがある。出口ウィンドウG30に表示されるクリップボードオブジェクトG34によって表されるクリップボードは、ルーム13の所属するオフィス14に付属するクリップボードであり、ユーザのルームウィンドウG01内に表示されるクリップボードオブジェクトG071によって表されるクリップボードは、そのルーム13に附属するクリップボードである。このようにクリップボード24はその管理の形態によって次の2つに分類される。

【0167】(1) ルーム13に付属するクリップボード24。

【0168】ルーム13のオブジェクトとして登録される個人用のクリップボード24で、その有効範囲は一利用者となる。また、前述のデータ18の公開操作と同様の処理を施すことにより、他のユーザからもアクセスできるようにすることで、クリップボード24の有効範囲を任意に指定された複数利用者の集合とすることもできる。

【0169】(2) オフィス14に付属するクリップボード24。

【0170】オフィス14のオブジェクトとして登録されクリップボード24で、そのオフィス17に含まれるルーム13のユーザのみか共通にアクセスできるクリップボード24であり、その有効範囲はシステムによって恒久的に指定された複数の集合となる。

【0171】＜クリップデータのクリップボードへの登録＞アプリケーション26が、デスクトップ管理システムのクリップデータ32を登録する操作および手順を図62のフローチャートを用いて説明する。

【0172】アプリケーション26は、すでに公知の任意のプログラム間通信手段を使用しデスクトップにメッセージを送る。このメッセージには、「クリップデータ32を登録する」というコマンドと、登録するクリップデータ32のフォーマットID113が含まれている。アプリケーション26からメッセージを受けとったデスクトップ管理システムは、S230でメッセージを解析し、アプリケーションにより指定された、クリップデータ32のフォーマットID113を得る。

【0173】次に、S231で利用者に対して新たに登録するクリップデータ32の名称を付けることを促すメッセージを表示し、クリップデータ32の名称を得る。

【0174】次に、S232でデータをクリップボードにコピー処理し、S233で、得られた名称と、アプリケーション26により指定されたフォーマットID113とともに、クリップデータ32をクリップボード24に登録し、図61で示すクリップボードウィンドウG62内にクリップデータオブジェクトG63として表示する。

【0175】＜クリップデータの選択＞アプリケーション26が、デスクトップ管理システムのクリップボード24に登録されているクリップデータ32を利用するための手順を、図63にしたがって説明する。

【0176】アプリケーション26が、デスクトップ管理システムのクリップボード24中に登録されているクリップデータ32を使用する場合、すでに公知の任意のプログラム間通信を使用してデスクトップ管理システムにメッセージを送る。このメッセージには「クリップデータ32の要求」というコマンドと、アプリケーション26で利用可能な一つまたは複数のフォーマットID113とが含まれている。

【0177】アプリケーション26からメッセージを受

けとったデスクトップ管理システムは、S240でメッセージを解析し、要求された一つまたは複数のフォーマットID113を得る。

【0178】次にS241で、クリップボード24に登録されているクリップデータ32の中から、アプリケーション26により指定された、フォーマットID113と一致するフォーマットIDを持つ、一つまたは複数のクリップデータ32をクリップボード24内から検索する。

【0179】次に、S242でクリップボードウィンドウG62をオープンして表示し、ユーザにクリップデータ32を選択することを促すメッセージとともに、先に検索を行なった、アプリケーション26により指定されたフォーマットID113と一致するフォーマットIDを持つ、クリップボード24内の一つまたは複数のクリップデータ32をクリップデータオブジェクトG63として表示する。

【0180】S243で、ユーザによってクリップデータ32が選択されると、S244でデスクトップ管理システムは選択されたクリップデータ32を前記通信手段を使用してアプリケーション26に通知する。

【0181】<カットバッファ>次に、デスクトップ管理システムの持つカット&コピー&ペーストバッファ（以下、「CCPバッファ」と呼称する）の機能について説明する。

【0182】アプリケーション26はデータ18またはその一部を、そのアプリケーション内、または他の一つまたは複数のアプリケーションに複写または移動させることがあり、そのために作成するデータ、およびそのデータを一時的に格納する場所をそれぞれ「CCPデータ」、「CCPバッファ」という概念で管理する。この「CCPバッファ」は前述のクリップボード24とは、格納した「CCPデータ」を表示する手段を持たない点において異なる。

【0183】図64はCCPデータ管理情報160を表す図である。CCPデータ管理情報160は、デスクトップ管理システムが「CCPデータ」を管理するための情報で、ルーム13毎に存在する。

【0184】CCPデータ管理情報160には一つの「CCPデータ」についての次の項目が含まれる。CCPデータの内容であるファイル名161、CCPデータのデータ種別を表すフォーマットID162、CCPデータのデータサイズ163、CCPデータを作成したユーザのユーザID164、CCPデータを作成したユーザのグループID165である。「CCPバッファ」にはフォーマットID162毎に1つの「CCPデータ」が格納されるようになっており、アプリケーション26がフォーマットID162を指定して「CCPバッファ」への登録を行なったとき、「CCPバッファ」にすでに指定されたフォーマットID162の「CCPデー

タ」が登録されている場合は、後から登録を行なったもののみが有効な「CCPデータ」として格納される。

【0185】アプリケーションはフォーマットID162を指定することで、「CCPバッファ」内に唯一存在するその指定のフォーマットID162の「CCPデータ」を得ることが出来る。前述のクリップボード24には、フォーマットID113毎に複数のクリップデータ32を格納することができたが、アプリケーション26がフォーマットID113を指定してクリップデータ32を要求した場合、そのフォーマットID113のクリップデータ32を示すクリップデータオブジェクトG63を、表示されたクリップボードウィンドウG62に表示してユーザに選択させる必要があった。しかし、この「CCPバッファ」では、アプリケーションによって指定されたフォーマットIDの「CCPデータ」は多くとも1つしか存在しないため、クリップボード24のようにユーザに選択させる必要がないこともクリップボードと異なる。「CCPバッファ」に格納される「CCPデータ」は、デスクトップ管理システムにより管理されているが、通常のキャビネット16内のデータとは異なり、ユーザが視覚的に見ることは出来ない。

【0186】「CCPデータ」はアプリケーション26により作成され、そのデータの種別を表すフォーマットID162がそのアプリケーションによって設定される。アプリケーション26により作成された「CCPデータ」は、そのアプリケーション26を操作しているユーザのルームに附属する「CCPバッファ」に格納される。「CCPバッファ」内の各「CCPデータ」はそれぞれ固有のフォーマットID162を持っており、アプリケーション26が特定のフォーマットID162をもつ「CCPデータ」をデスクトップ管理システムに対して要求することにより、デスクトップ管理システムは、そのユーザのルーム13に附属する「CCPバッファ」内の、フォーマットID162を持つ「CCPデータ」を特定し、アプリケーションに渡すことが出来る。次に、図にしたがって説明する。

【0187】<CCPバッファへの登録>アプリケーション26が、デスクトップ管理システムの管理する「CCPバッファ」に「CCPデータ」を登録する手順を、図65にしたがって説明する。

【0188】アプリケーション26は、すでに公知の任意のプログラム間通信手段を使用しデスクトップ管理システムにメッセージを送る。このメッセージには、「CCPデータを登録する」というコマンドと、登録をおこなうアプリケーションが指定する「CCPデータ」のフォーマットID162とが含まれる。

【0189】アプリケーション26からメッセージを受けとったデスクトップ管理システムは、S250でメッセージを解析し、アプリケーション26により指定された「CCPデータ」のフォーマットID162を得る。

【0190】次に、S251でアプリケーション26により指定されたフォーマットID162とともに、「CCPデータ」を「CCPバッファ」に登録する。

【0191】<CCPデータの取得>アプリケーション26が、デスクトップ管理システムの管理する「CCPバッファ」に登録されている「CCPデータ」を利用するための手順を、図66にしたがって説明する。

【0192】アプリケーション26が、デスクトップ管理システムの「CCPバッファ」の「CCPデータ」を使用する場合、すでに公知の任意のプログラム間通信を使用し、デスクトップにメッセージを送る。このメッセージには、「CCPデータの要求」というコマンドと、「CCPデータ」を特定するためのフォーマットID162が含まれる。アプリケーション26からメッセージを受けとったデスクトップ管理システムは、S260でメッセージを解析し、要求されたフォーマットID162を得る。

【0193】次に、S261で受けとったフォーマットID162に一致する「CCPデータ」の検索を行なう。検索の結果、アプリケーション26により指定された特定のフォーマットID162を持つ「CCPデータ」が存在するかをS262で判断し、アプリケーション26により指定された特定のフォーマットID162を持つ「CCPデータ」が見つからなかった場合には、そのまま終了する。

【0194】S262の判断の結果、アプリケーション26によりしてされた特定のフォーマットID162を持つ「CCPデータ」を得られた場合には、S263で「CCPバッファ」内の唯一の「CCPデータ」を前記通信手段を使用してアプリケーション26に通知する。<検索>検索は、例えばキャビネット16下のバインダ20、ライブラリバインダ27、データ18を検索対象とし、これらに対して検索条件を設定し、検索条件に合致するバインダ20、ライブラリバインダ27、データ18をウィンドウに表示等する。

【0195】検索条件とは、例えば検索対象であるバインダ20、ライブラリバインダ27、データ18に付帯する情報で、本実施例では図20に示したオブジェクト管理情報110の登録日付117、更新日付118、漢字名称115、所有者ユーザID120、所有者グループID121、オブジェクトタイプ111などの各項目と、およびこれらの論理的な組合せである。検索条件の方法は、例えば取り得る検索条件を選択肢として全て利用者に提示し、その中から選択する形式であってもよいし、利用者に逐一入力させる形式であってもよい。検索結果を利用者に提示する方法については、例えば、

(あ) 検索処理を開始して終了するまでの特定期間だけ特定領域を検索結果表示に充当し、専用の検索結果表示手段を持たない方法。

【0196】(い) 検索条件に合致したものをだけを表示

するための専用のウィンドウなどを新たに生成する、検索結果表示手段を持つ方法。

【0197】(う) 必要に応じて(あ)、(い)の両者を組み合わせる方法。などが考えられる。

【0198】次に、検索処理とその検索結果を利用者に提示する方法の実施例について説明する。

【0199】はじめに、キャビネット16直下の階層に存在するバインダ20の検索処理について、上記の検索結果を利用者に提示する方法のそれぞれについて説明する。本実施例においては検索処理に際してキャビネット16のオブジェクト管理情報110を利用する方法について記述するが、検索処理用のデータ構造を作成して行う実施例も考えられる。また、説明の便宜上、本実施例では検索情報として、オブジェクト管理情報110のオブジェクトタイプ111の内容など単純な条件を設定するが、前述のように検索条件の設定にはこだわらない。

【0200】前記(あ)の実施例を、図6、図67、フローチャート図68を使って説明する。検索結果表示には、キャビネット16内オブジェクトを一覧表示するウィンドウも充当するものとする。

【0201】初め、全てのキャビネット16内オブジェクトが図5に示すようにキャビネットウィンドウG16内に一覧表示されている。これらに対して、「キャビネット16内のバインダ20を検索する」として検索処理を実行すると、検索条件としてキャビネット16のオブジェクト管理情報110のオブジェクトタイプ111がバインダであるかどうかを検査され(S300)、キャビネットウィンドウG16から合致しないものの表示を消去する(S301)。この処理をキャビネット16の全エントリ(オブジェクト)に対して行って(S302)検索結果のユーザへの提示が終了する。

【0202】この結果、図67に示すように、キャビネットウィンドウG100内には、検索条件に合致したバインダ20を示すバインダオブジェクトG101だけが残留表示され、合致しなかったもの(即ちデータ18)は非表示状態となる。

【0203】次に、(い)の実施例について、図5、図69、フローチャート図70を使って説明する。初め、全てのキャビネット16内オブジェクトが図5に示すようにキャビネットウィンドウG16内に一覧表示されている。これらに対して、「キャビネット16内のバインダ20を検索する」として検索処理を実行すると、図69に示すような検索結果表示ウィンドウG110を生成し(S310)、検索条件としてキャビネット16のオブジェクト管理情報110のオブジェクトタイプ111がバインダであるかどうかを検査され(S311)、合致するものを検索結果表示ウィンドウG110に検索結果オブジェクトG111として表示する(S312)。

この処理をキャビネット16の全エントリ(オブジェクト)に対して行って(S313)検索結果のユーザへの

提示が終了する。

【0204】この結果、キャビネットウィンドウG16の表示は不変だが、図69に示すように、検索結果表示ウィンドウG110には条件に合致したバインダ20だけが新たに検索結果オブジェクトG111として表示される。

【0205】次に、(う)の実施例について図5、図67、図71、フローチャート図73を使って説明する。

【0206】初め、全てのキャビネット16内オブジェクトが図5に示すようにキャビネットウィンドウG16内に一覧表示されている。これらに対して、「キャビネット16内のバインダ20を検索する」として検索処理を実行すると、図71に示すような検索結果詳細ウィンドウG120を生成し(S330)、検索条件としてキャビネット16のオブジェクト管理情報110のオブジェクトタイプ111がバインダであるかどうかを検査し(S331)、キャビネットウィンドウG16から合致しないものの表示を消去する(S332)。また、条件に合致したものの付帯情報を検索結果詳細ウィンドウG120に表示する(S333)。この処理をキャビネット16の全エントリ(オブジェクト)に対して行って(S334)検索結果のユーザへの提示が終了する。

【0207】この結果、キャビネットウィンドウG16内には、図67に示すように検索条件に合致したバインダ20を示すバインダオブジェクトG17だけが残留表示され、合致しなかったもの(即ちデータ18)は非表示状態となる。さらに、検索結果詳細ウィンドウG120には図71に示すように、キャビネットウィンドウG16に残留表示されたバインダ20に付帯する詳細情報が表示される。

【0208】以上述べたキャビネット16直下の階層に存在するバインダ20の検索処理手順とほぼ同一の処理手順によって、キャビネット16直下の階層にあるデータ18の検索、キャビネット16直下の階層にあるライブラリバインダ27の検索、キャビネット16直下の階層にあるライブラリデータ28の検索、バインダ20直下の階層にあるバインダ20の検索、バインダ20直下の階層にあるデータ18の検索、ライブラリバインダ27直下の階層にあるライブラリバインダ27の検索、ライブラリバインダ27直下の階層にあるデータ28の検索、公開棚22直下の階層にある公開トレイ29の検索、タグトレイ19直下の階層にあるデータ18やバインダ20などの検索などが、オブジェクトタイプ111を検索条件として設定することによって実現可能である。

【0209】以下の検索処理の実施例では、説明の便宜上、(い)の検索結果を表示する専用の検索結果ウィンドウG110を生成する実施例に従って説明する。

【0210】次に、キャビネット16から2階層下に存在するオブジェクトの検索処理についてフローチャート

図72を使いながら説明する。

【0211】キャビネット16の2階層下に存在するオブジェクトとは、例えばキャビネット16の直下階層にバインダ20が存在し、そのバインダ20の直下階層に存在するオブジェクトを指して言う。「キャビネット16から2階層下に存在するデータ」を検索条件として設定され検索が実施されると、検索結果を表示するための検索結果ウィンドウG110を生成する(S320)。キャビネット16のオブジェクト管理情報110からオブジェクトタイプ111がバインダであるエントリを選ぶ(S321)。

【0212】このエントリの情報を元にオブジェクト管理情報110を読み込んでオブジェクトタイプ111がデータ18であるかどうかを検査し(S322)、合致するエントリのフォーマットID113とデータ情報50からアイコンファイルを読み出して検索結果ウィンドウG110に検索結果オブジェクトG111として表示し(S324)、この処理を該バインダ20のオブジェクト管理情報110の全エントリに対して行って(S325)、キャビネット16のオブジェクト管理情報110からオブジェクトタイプ111がバインダ20であるエントリを選ぶ処理(S321)へ戻る。

【0213】この処理をキャビネット16のオブジェクト管理情報110の全エントリに対して行って(S326)検索処理を終了する。

【0214】このような処理を再帰的に行うことにより階層数にかかわらず検索が実施できる。また、異なる階層に存在するオブジェクトを検索対象として検索条件が設定され検索が実施された場合、検索処理の再帰的な実行回数を変更することにより実現可能である。これをフローチャート図74を使って説明する。この例では、階層にかかわらずキャビネット16の下方階層にあるオブジェクトを検索対象として、「オブジェクト管理情報110のオブジェクトタイプ111がデータでかつクリエイタID114が012である」ことを検索条件として検索を実施するものとする。また、本実施例においては、説明の便宜上データ18が存在し得るのはキャビネット16直下あるいはバインダ20直下であるとする。

【0215】キャビネット16から任意階層下方に存在するオブジェクトを検索対象として検索条件が設定され検索が実施されると、検索結果を表示するための検索結果ウィンドウG110を生成する(S340)。次にはキャビネット16を検索対象として、検索条件を渡して定型処理である「処理」を実行して終了する(S341)。

【0216】次に、フローチャート図75を使って図74のステップS341の「処理」について説明する。

「処理」はキャビネット16に限らずバインダ20、ライブラリバインダ27、公開棚22、公開トレイ29、タグトレイ19などのように、更なる下方階層を持ち得

るオブジェクトに対して、再帰的にデータ18を検索抽出し、検索条件に合致するデータ18の情報を検索結果ウィンドウG110に表示する機能を実現するモジュールである。

【0217】「処理」が呼び出されると、呼び出し時に指定されたオブジェクトのオブジェクト管理情報110からオブジェクトタイプ111がデータ18であるエントリを選び（S350）、更にそのエントリに対して検索条件「オブジェクト管理情報110のオブジェクトタイプ111がデータでかつクリエイタID114が012である」と合致するかを判定し（S351）、合致するエントリのフォーマットID113とデータ情報50からアイコンファイルを読み出して検索結果ウィンドウG110に必要な情報とともに検索結果オブジェクトG111として表示する（S352）。

【0218】一方、呼び出し時に指定されたオブジェクトのオブジェクト管理情報110からオブジェクトタイプ111がバイндаであるエントリに対しては、再帰的に検索条件、検索対象とともに「処理」を呼び出す（S354）。これらの処理を呼出時に指定されたオブジェクトのオブジェクト管理情報110の全エントリに対して行って（S353）、「処理」からリターンする。

【0219】＜検索結果への到達＞次に検索結果を元に、データの管理構造をたどることなく検索条件に合致したオブジェクトに到達する処理について説明する。本実施例において、検索対象となっているのはキャビネット16下のデータ18およびバイнда20とする。はじめに、検索結果にしたがって、データ管理構造をたどることなく検索条件に合致したオブジェクトに到達するための情報を生成する処理について図79を使って説明する。

【0220】検索処理が実行されると、検索結果ウィンドウG110を生成し（S360）、同時にこの検索結果ウィンドウG110がキャビネット16やバイнда20のような、その下位に階層構造を持つオブジェクトとして、検索結果ウィンドウG110内の検索結果オブジェクトについての情報を格納するオブジェクト管理情報110を生成する（S361）。検索処理の実行にともない検索条件に合致した（S362）ものは検索結果ウィンドウG110に表示し（S363）、同時に検索条件に合致することを判断するために使用した、検索対象キャビネット16のオブジェクト管理情報110の該当するエントリ情報を、検索結果ウィンドウG110のオブジェクト管理情報のエントリにコピーする（S364）。この処理を全検索対象に関して実行する（S365）。

【0221】こうして構成された検索結果ウィンドウG110に対して、オープン操作が行われた際の処理について図77を使って説明する。検索結果ウィンドウG110に表示されているのは、検索条件に合致したバイ

ダ20およびデータ18を示す検索結果オブジェクトG111であり、検索結果ウィンドウG110内の検索結果オブジェクトG111に対する操作は、実体であるバイнда20およびデータ18を示すそれぞれバイндаオブジェクトG17およびデータオブジェクトG18に対する操作と同じ処理が行なわれる。

【0222】すなわち、検索結果表示ウィンドウG110上の検索結果オブジェクトG111に対してオープン操作が行われると、検索結果ウィンドウG110のデータ管理情報110から、指定されたオブジェクトに該当するエントリの情報を得て（S370）、指定されたオブジェクトの種別にしたがって（S371）バイнда22またはデータ18のオープン処理が呼ばれる（S372、S373）。検索結果に対する削除操作に関しては、これを禁止することも、検索結果ウィンドウG110上の、検索結果オブジェクトの示すデータ18またはバイнда20への到達指標だけを削除して実体であるバイнда20あるいはデータ18は削除しないように設定することも可能である。

【0223】＜階層構造を考慮した利用者への検索結果の提示＞次にデータ18とバイнда20のように、階層構造をなすオブジェクトが同時に検索条件に合致した際の、階層構造を考慮した利用者への検索結果提示の実施例について、フローチャート図78と図2、図79、図80を使って説明する。本実施例においては、キャビネット16から任意階層下方に存在するデータ18とバイнда20を検索対象として検索条件が設定され、すでに説明した検索処理が実施され、その結果、図79（A）に太枠で示すような階層構造にあるデータ18とバイнда20とが検索条件に合致したと仮定する。また、検索結果は実際のデータの階層構造に無関係に、図79

（B）のように検索結果ウィンドウG110に検索結果オブジェクトとして表示されているとする。

【0224】ここで検索結果ウィンドウG110から、名称が「バイнда0」であるバイнда20を示す検索結果オブジェクトをオープンすると、すでに説明したように検索結果オブジェクトに対する操作は、それが示すバイнда20に対する操作としてバイнда20がオープンされ、バイндаウィンドウG22が表示される。このバイндаウィンドウG22の表示においては、検索結果ウィンドウG110内に表示された検索結果オブジェクトG111の示すバイнда20またはデータ18を示すバイндаオブジェクトG20またはデータオブジェクトG18は、先の検索で検索条件に該当したオブジェクトであることを明示するために図80のようにアイコンが網掛けになるなどの表示が変更される。このような表示を行うための処理についてフローチャート図78を使って説明する。

【0225】図79の名称「バイнда0」のバイнда20を示す検索結果オブジェクトG130が2クリックさ

れてオープンされると、バインダウィンドウG22を生成し(S380)、検索結果オブジェクトG130のバインダ20のオブジェクト管理情報110の各エントリのフォーマットID113とデータ情報36からアイコンファイルを読み出して前記ウィンドウに必要な情報とともにオブジェクトの表示を行う(S381)。次にこのエントリと同じ情報が、検索結果ウィンドウG110のオブジェクト管理情報110のエントリに含まれていないかを調べ(S382)、含まれていた場合、前記ウィンドウへのアイコン表示に、例えば図80に示すような網掛け処理を重畳して行い、ユーザの識別を容易にする(S383)。以上の処理を名称「バインダ0」のバインダ20のオブジェクト管理情報110の全エントリに対して行って(S384)終了する。

【0226】次に検索結果に対してアプリケーションプログラムを起動して、任意の処理を行う実施例についてフローチャート図81と図82を使って説明する。図82は、検索結果ウィンドウG110であり、表示されているのは検索条件に合致したデータ18示す検索結果オブジェクトG111である。この検索結果に対して実行する処理は、編集、印刷など多数考えられるが、本実施例においては印刷を実行するアプリケーションプログラムを起動して印刷処理するものとする。処理を行うアプリケーションプログラムの起動は、例えばリストから処理を選択する形式で行う。

【0227】データ18を示す検索結果オブジェクトG150が選択され、検索結果ウィンドウG110上の操作ボタンG140が選択される(S390)と、検索結果ウィンドウG110のオブジェクト管理情報110の、選択された検索結果オブジェクトG150に該当するデータ18のエントリから、選択されているデータ18のオフID51とクリエイタID114を得る(S391)。これとアプリケーション情報60を比較し、クリエイタID114と同じアプリケーションID61を持つエントリ中の、そのフォーマットID51に該当する処理リスト65の処理名称を得て(S392)、メニューリストG141として表示する(S393)。

【0228】ユーザによってリストG141の1項目が選択されると(S394)、検索結果表示ウィンドウG110のオブジェクト管理情報110のエントリからファイル名112を得る(S395)。このファイル名112と、既に得られたフォーマットID51と、リストG141から選択された項目に対応する処理コードとをアプリケーションプログラムに渡して、アプリケーションID61に対応する実行モジュール名62によってアプリケーションを起動する(S396)。複数のデータが選択された場合、上記一連の処理を繰り返すことで実施が可能である。

【0229】＜公開情報を考慮した検索処理の実行＞次に、他のユーザが所有するオブジェクトに対する検索が

実行された場合の処理についてフローチャート図83を使って説明する。本実施例においては、ユーザAがユーザBのキャビネット16以下のデータ18と、ユーザCのキャビネット16以下のデータ18を検索対象とし、検索条件を設定して検索処理を実行するものとする。また、すでに説明した公開操作に従った方法によって、ユーザBのキャビネット16は検索処理の実行ユーザであるユーザAに公開されており(この状態を、「ユーザAは該オブジェクトに関してユーザBの公開ユーザである」と称する)、ユーザCのキャビネット16はユーザAに対して一切公開されていない(この状態を、「ユーザAは該オブジェクトに関してユーザCの非公開ユーザである」と称する)ものとする。

【0230】ユーザAが、ユーザB、Cのキャビネット16を検索対象として検索条件を設定して検索処理を実行すると、検索結果表示ウィンドウG110を生成する(S400)。他ユーザのキャビネット16などを検索する際には、公開トレイ29のオブジェクト管理情報110から、検索しようとするキャビネット16に該当するエントリを探し(S401)、存在する場合は該キャビネット16に対して、フローチャート図75に示すところの「処理」を、検索条件とともに呼び出す(S402)。該当するエントリが存在しない場合は、ユーザが検索対象として指定したオブジェクトが利用者Aに公開されていないことを表示する(S403)。これを、この処理検索対象として指定された、他ユーザが所有する全てのオブジェクトに対して繰り返して(S404)、他のユーザが所有するオブジェクトに対する検索処理を終了する。

【0231】次に、他のユーザが所有するオブジェクトに対する検索が実行された場合に、検索処理は実行した上で、公開情報に従ってユーザへの検索結果の提示を抑制する処理についてフローチャート図84、図85を使って説明する。本実施例においては、ユーザAがユーザBのキャビネット16以下のデータ18と、ユーザCのキャビネット16以下のデータ18を検索対象とし、検索条件を設定して検索処理を実行するものとする。また、＜公開操作＞に従った方法によって、ユーザBのキャビネット16は検索処理の実行ユーザであるユーザAに公開されており、ユーザCのキャビネット16はユーザAに対して一切公開されていないものとする。

【0232】ユーザAが、ユーザB、Cのキャビネット16を検索対象として検索条件を設定して検索処理を実行すると、検索結果表示ウィンドウG110を生成する(S410)。

【0233】次に、指定されたキャビネット16に対して、後述する「表示を伴わない処理」を、検索条件とともに呼び出す(S411)。次には、公開トレイ29のオブジェクト管理情報110から、検索したキャビネット16に該当するエントリを探し(S412)、存在す

る場合は「表示を伴わない処理」が返した情報を元に、検索結果表示ウィンドウG110に検索結果の表示を行う(S413)。該当するエントリが存在しない場合は、「表示を伴わない処理」が返した情報から、非公開ユーザグループユーザに知られても指し支えない情報や、非公開ユーザグループユーザであってもシステムの管理などの理由で知る必要性が認められる情報などを選択的に検索結果表示ウィンドウG110に表示する(S414)。この処理を検索対象として指定された、他ユーザが所有する全てのオブジェクトに対して繰り返して(S415)、他のユーザが所有するオブジェクトに対する検索処理を終了する。

【0234】前記の非公開ユーザに知られても指し支えない情報とは、例えば検索条件に合致するオブジェクトの存在(存在する/しない)等である。

【0235】また、前記非公開ユーザであってもシステムの管理などの理由で知る必要性が認められる情報とは、オブジェクトの名称など、オブジェクトを特定し得る情報を表示しないで表示する、オブジェクトのサイズや登録日付、更新日付などである。

【0236】次に、フローチャート図85を使って、図84のステップS411の「表示を伴わない処理」について説明する。「表示を伴わない処理」はキャビネット16に限らずバインダ20、ライブラリバインダ27、公開棚22、公開トレイ29、タグトレイ19などのように、更なる下方階層を持ち得るオブジェクトに対して、再帰的にデータ18を検索抽出し、検索条件に合致するデータ18の情報を検索結果表示ウィンドウG110に表示する機能を実現するモジュールである。

【0237】「表示をもとなわかない処理」が呼び出されると、リターン情報領域を確保して初期化し(S420)、呼び出し時に指定されたオブジェクトのオブジェクト管理情報110からオブジェクトタイプ111がデータ18であるエントリを選ぶ(S421)。更にそのエントリに対して検索条件と合致するかを判定し(S422)、合致するエントリの情報をリターン情報としてセットする(S423)。一方、呼び出し時に指定されたオブジェクトのオブジェクト管理情報110からオブジェクトタイプ111がデータ18以外であるエントリに対しては、再帰的に検索条件、検索対象とともに「表示を伴わない処理」を呼び出す(S424)。これらの処理を呼出時に指定されたオブジェクトのオブジェクト管理情報110の全エントリに対して行って(S425)、リターン情報を「表示を伴わない処理」からリターンする。

【0238】＜ユーザグループへの公開情報を考慮した検索処理の実行＞次に他のユーザが所有するオブジェクトに対する検索が実行された場合の処理についてフローチャート図86を使って説明する。本実施例においては、利用者AがユーザBのキャビネット16以下のデー

タ18と、ユーザCのキャビネット16以下のデータ18を検索対象とし、検索条件を設定して検索処理を実行するものとする。また、＜公開操作＞に従った方法によって、ユーザBのキャビネット16は検索処理の実行ユーザであるユーザAが属するユーザグループに公開されており(この状態を、「ユーザAは該オブジェクトに関してユーザBの公開ユーザグループユーザである」と称する)、ユーザCのキャビネット16はユーザAが属するユーザグループに対して一切公開されていない(この状態を、「ユーザAは該オブジェクトに関してユーザCの非公開ユーザグループユーザである」と称する)ものとする。

【0239】ユーザAが、ユーザB、Cのキャビネット16を検索対象として検索条件を設定して検索処理を実行すると、検索結果表示ウィンドウG110を生成する(S430)。他ユーザのキャビネット16などを検索する際には、公開トレイ29のオブジェクト管理情報110から、検索しようとするキャビネット16に該当するエントリを探し(S431)、存在する場合は該キャビネット16に対して、フローチャート図75に示すところの「処理」を、検索条件とともに呼び出す(S432)。該当するエントリが存在しない場合は、ユーザAが検索対象として指定したオブジェクトが利用者Aが属するユーザグループに公開されていないことを表示する(S433)。この処理を検索対象として指定された、他ユーザが所有する全てのオブジェクトに対して繰り返して(S434)、他のユーザが所有するオブジェクトに対する検索処理を終了する。

【0240】次に他のユーザが所有するオブジェクトに対する検索が実行された場合に、検索処理は実行した上で、公開情報に従ってユーザへの検索結果の提示を抑止する処理についてフローチャート図87、図2を使って説明する。

【0241】本実施例においては、ユーザAがユーザBのキャビネット16以下のデータ18と、ユーザCのキャビネット16以下のデータ18を検索対象とし、検索条件を設定して検索処理を実行するものとする。また、＜公開操作＞に従った方法によって、ユーザBのキャビネット16は検索処理の実行ユーザであるユーザAが属するユーザグループに公開されており、ユーザCのキャビネット16はユーザAが属するユーザグループに公開されており、ユーザCのキャビネット16はユーザAが属するユーザグループに対して一切公開されていないものとする。

【0242】ユーザAが、ユーザB、Cのキャビネット16を検索対象として検索条件を設定して検索処理を実行すると、検索結果表示ウィンドウG110を生成する(S440)。次に、指定されたキャビネット16に対して、＜公開情報を考慮した検索処理の実行＞で述べた図85の「表示を伴わない処理」を、検索条件とともに



呼び出す（S441）。次には、公開トレイ29のオブジェクト管理情報110から、検索したキャビネット16に該当するエントリを探し（S442）、存在する場合は「表示を伴わない処理」か返した情報を元に、検索結果表示ウィンドウG110に検索結果の表示を行う（S443）。該当するエントリが存在しない場合は、「表示を伴わない処理」が返した情報から非公開ユーザに知られても指し支えない情報や、非公開ユーザであってもシステムの管理などの理由で知る必要性が認められる情報などを選択的に検索結果表示ウィンドウG110に表示する（S444）。この処理を検索対象として指定された、他ユーザが所有する全てのオブジェクトに対して繰り返して（S445）、他のユーザが所有するオブジェクトに対する検索処理を終了する。

【0243】以上説明したように、指定された利用者のデータを処理可能なアプリケーションを示すアイコンをハイライト表示や、線で結んだ表示をすることにより、また、両アイコンの距離を近づける或は遠ざけるといった処理により、アプリケーションで参照、更新できる利用者データの対応関係を容易に利用者に提示し、容易なアプリケーションの起動方法を提供することにより、利用者がデータとアプリケーションの対応づけを意識することなく、また、アプリケーションの起動方法を知らなくとも、データの参照、更新が可能となる。

【0244】また、複数の利用者を持つ複数の計算機システム上で、分散している任意のデータを任意の利用者が、データが管理されている構造や計算機システムを意識することなく、簡単に検索することが可能であり、検索されたデータを処理することが容易である。

【0245】また、複数の利用者を持つ複数の計算機システム上で、分散している任意のデータに対する任意の一利用者の検索処理を、或は検索処理の結果の表示を制限的に実行することが可能となる。さらに利用者管理手段、第二の操作、管理手段、第二の検索実行制御手段を備えることにより、複数の利用者を持つ複数の計算機システム上で、分散している任意のデータに対する一利用者の検索処理、或は検索処理結果の表示を制限的に実行することが可能となる。

【0246】尚、本発明は、複数の機器から構成されるシステムに適用しても、1つの機器から成る装置に適用しても良い。また、本発明はシステム或は装置にプログラムを供給することによって達成される場合にも適用できることは言うまでもない。

#### 【0247】

【発明の効果】以上説明したように、本発明に係るデータ処理装置によれば、データがどこに格納されていても利用者がそれを意識することなく、所望のデータを操作することができる。また、データとそれを処理することのできるアプリケーションプログラムとの対応を利用者が意識しなくともよいという効果がある。

#### 【図面の簡単な説明】

【図1】実施例のデータ処理装置のブロック図である。

【図2】デスクトップ管理システムにおけるルームとオフィスの説明図である。

【図3】ユーザのルームウインドウの表示例である。

【図4】デスクトップウインドウの表示例である。

【図5】キャビネットウインドウの表示例である。

【図6】ライブラリウインドウの表示例である。

【図7】ライブラリバインダウインドウの表示例である。

【図8】タグトレイウインドウの表示例である。

【図9】公開棚ウインドウの表示例である。

【図10】公開ウインドウの表示例である。

【図11】出口ウインドウの表示例である。

【図12】オフィスに含まれる出口の表示例である。

【図13】ルームに含まれる出口の表示例である。

【図14】データ情報の一例である。

【図15】アプリケーション情報の一例である。

【図16】ライブラリ情報の一例である。

【図17】ユーザ情報の一例である。

【図18】オフィス情報の一例である。

【図19】グループ情報の一例である。

【図20】オブジェクト管理情報の一例である。

【図21】オープンオブジェクト管理情報の一例である。

【図22】ライブラリ検索順序情報の一例である。

【図23】キャビネット内のバインダをオープンする操作例である。

【図24】バインダをオープンする処理のフローチャートである。

【図25】キャビネットとバインダの間のデータコピーの操作例である。

【図26】キャビネットとバインダの間のデータコピー処理のフローチャートである。

【図27】アプリケーションオブジェクトをオープンする操作例である。

【図28】アプリケーションプログラムの起動処理のフローチャートである。

【図29】アプリケーションオブジェクトをオープンする操作例である。

【図30】アプリケーションプログラムの起動処理のフローチャートである。

【図31】アプリケーションオブジェクトをオープンする操作例である。

【図32】データオブジェクトをオープンする操作例である。

【図33】アプリケーションプログラムの起動処理のフローチャートである。

【図34】データオブジェクトをオープンする操作例である。

【図35】アプリケーションプログラムの起動処理のフローチャートである。

【図36】キャビネットとデスクトップ間のアプリケーションオブジェクトのコピー操作例である。

【図37】アプリケーションオブジェクト移動処理のフローチャートである。

【図38】キャビネットからデスクトップへのデータオブジェクトのドラッグ操作例である。

【図39】アプリケーションオブジェクトとデータオブジェクトとを結んで表示した例である。

【図40】アプリケーションオブジェクトとデータオブジェクトとを結んで表示する処理のフローチャートである。

【図41】アプリケーションオブジェクトとデータオブジェクトとの関係をハイライトで表示した例である。

【図42】アプリケーションオブジェクトとデータオブジェクトとの関係をハイライトで表示する処理のフローチャートである。

【図43】引力によるガイダンスの表示例である。

【図44】引力によるガイダンスでアプリケーションを起動する処理のフローチャートである。

【図45】反力によるガイダンスの表示例である。

【図46】公開トレイ作成操作例である。

【図47】公開設定操作例である。

【図48】公開トレイ作成処理のフローチャートである。

【図49】公開操作処理のフローチャートである。

【図50】公開設定処理のフローチャートである。

【図51】出口オープン処理のフローチャートである。

【図52】タグ作成処理のフローチャートである。

【図53】タグオープン処理のフローチャートである。

【図54】データ廃棄操作の表示例である。

【図55】データ廃棄処理のフローチャートである。

【図56】データ廃棄監視プロセスの処理のフローチャートである。

【図57】ライブラリバインダ作成操作の例である。

【図58】ライブラリバインダ処理のフローチャートである。

【図59】ライブラリデータ登録処理のフローチャートである。

【図60】ライブラリデータ検索処理のフローチャートである。

【図61】クリップボードウィンドウの表示例である。

【図62】クリップデータ登録処理のフローチャートである。

【図63】クリップデータ検索処理のフローチャートである。

【図64】CCPデータ管理情報の一例である。

【図65】CCPバッファへのデータ登録処理のフローチャートである。

【図66】CCPバッファのデータ検索処理のフローチャートである。

【図67】検索結果の表示例である。

【図68】検索処理のフローチャートである。

【図69】検索結果の表示例である。

【図70】検索処理のフローチャートである。

【図71】検索結果の詳細な表示例である。

【図72】検索処理のフローチャートである。

【図73】検索処理のフローチャートである。

10 【図74】階層に依存しない検索処理のフローチャートである。

【図75】階層に依存しない検索処理のフローチャートである。

【図76】検索処理のフローチャートである。

【図77】検索結果ウィンドウのオープン処理のフローチャートである。

【図78】検索結果の表示処理のフローチャートである。

【図79】検索操作対象の一例である。

20 【図80】検索結果の表示例である。

【図81】検索結果についてアプリケーションを起動する処理のフローチャートである。

【図82】検索結果についてアプリケーションを起動する操作の一例である。

【図83】検索処理のフローチャートである。

【図84】公開情報を考慮した検索結果の表示処理のフローチャートである。

【図85】公開情報を考慮した検索結果の表示処理のフローチャートである。

30 【図86】公開情報を考慮した検索処理のフローチャートである。

【図87】公開情報を考慮した検索結果の表示処理のフローチャートである。

【図88】反力によるガイダンスでアプリケーションを起動する処理のフローチャートである。

【符号の説明】

1…CRT、

2…ビデオRAM (VRAM)、

3…ビット処理ユニット (BMU)、

40 4…キーボード、

5…ポインティングデバイス (PD)、

6…CPU、

7…ROM、

8…RAM、

9…ハードディスクドライブ (HDD)、

10…フロッピーディスクドライブ (FDD)、

11…ネットワークインターフェース (Net-I/F)、

12…I/Oバス、

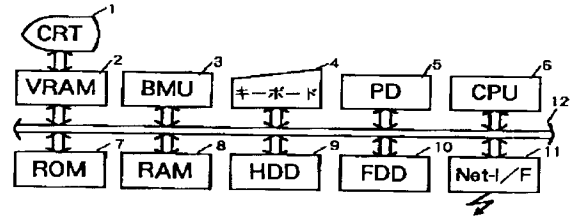
50 13…ルーム、

51

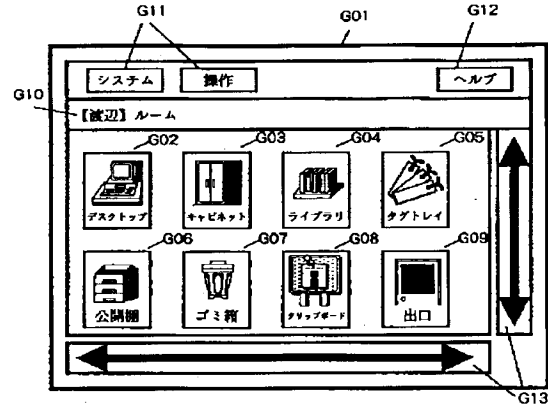
52

- 14…オフィス、
- 15…デスクトップ、
- 16…キャビネット、
- 17…ライブラリ、
- 18…データ、
- 19…タグトレイ、
- 20…バインダ、
- 21…タグ、
- 22…公開棚、
- 23…ごみ箱、
- 24…クリップボード、
- 25…出口、
- 26…アプリケーション、

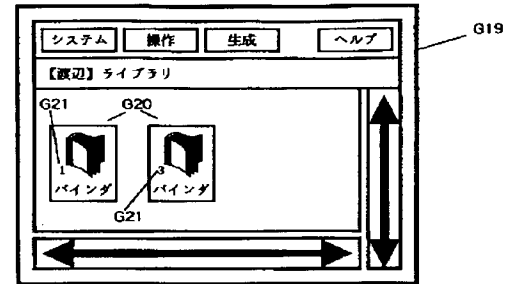
【図1】



【図3】

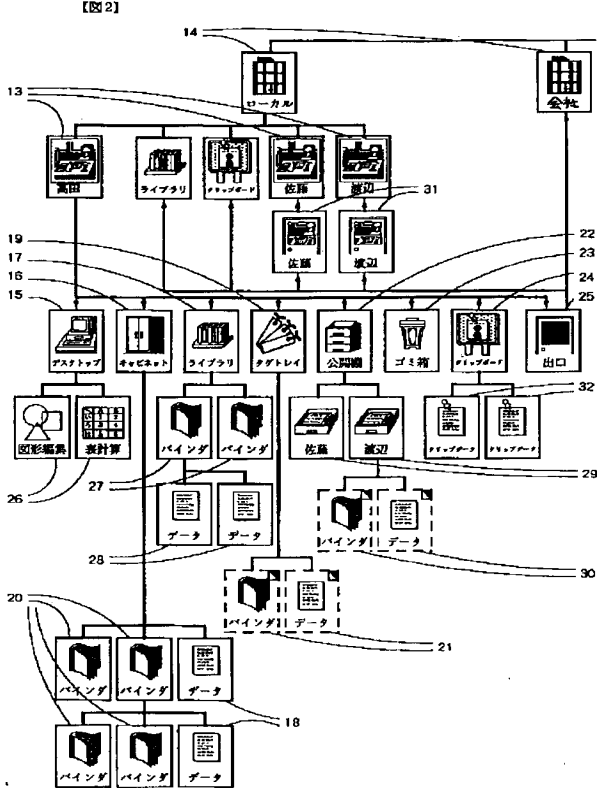


【図6】



- 27…ライブラリバインダ、
- 28…ライブラリデータ、
- 29…公開トレイ、
- 30…公開オブジェクト、
- 31…ドア、
- 32…クリップデータ、
- 50…データ情報、
- 60…アプリケーション情報、
- 70…ライブラリ情報、
- 80…ユーザ情報、
- 90…オフィス情報、
- 100…グループ情報、
- 110…オブジェクト管理情報である。

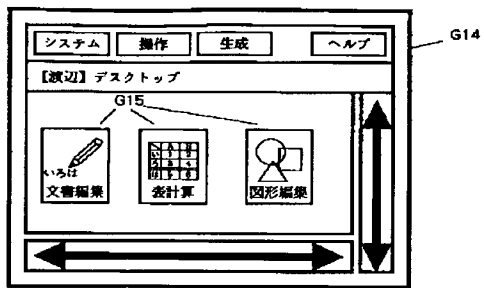
【図2】



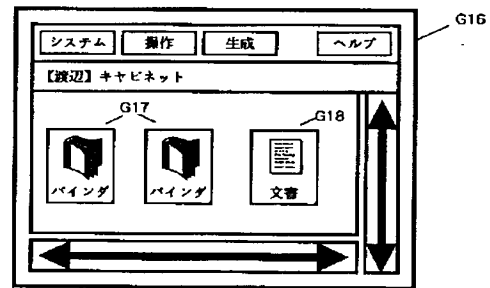
【図19】

グループID	ユーザID
100001	1001、1002、1003
100002	1001

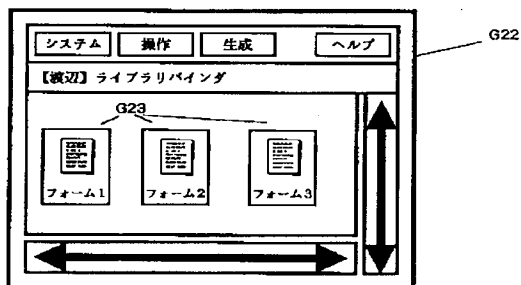
【図4】



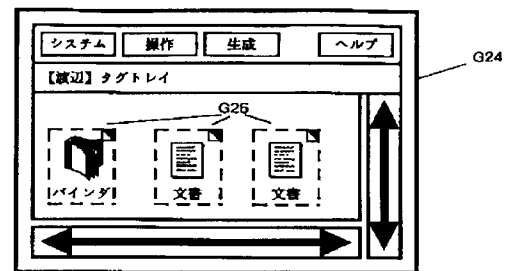
【図5】



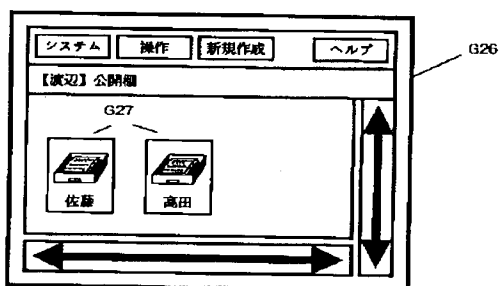
【図7】



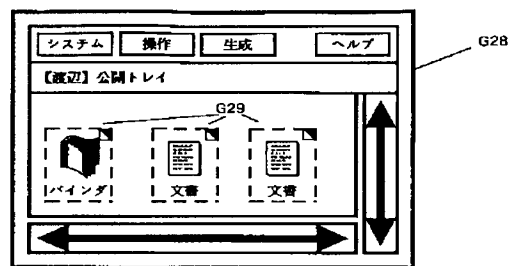
【図8】



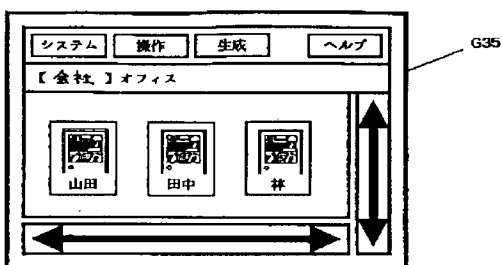
【図9】



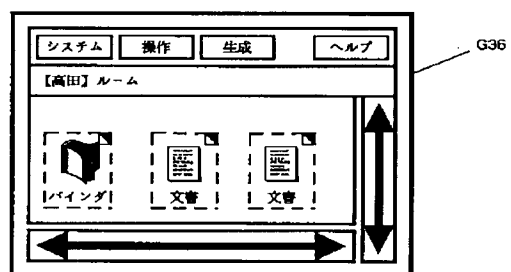
【図10】



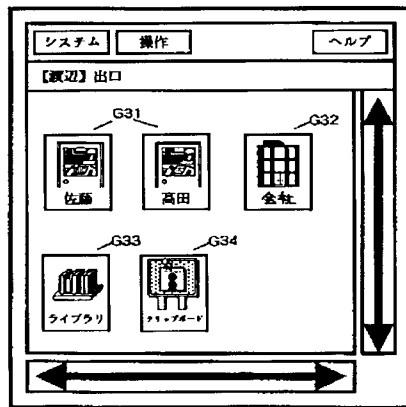
【図12】



【図13】



【図11】



【図14】

51 フォーマット ID	52 拡張子	53 アイコン ファイル名	54 ライブラリ ID
012	.etx	Eztext	020
013	.etb	Eztable	
212	.slk	Silk	

【図15】

61 アプリ ケーション ID	62 実行 モジュール名	63 アプリ ケーション 名称	64 アイコン ファイル名	65 処理リスト			60 処理 データ 必要
				フォーマット ID	処理 コード	処理名称	
012	ezps	文書編集	Ezps	012	e	編集	
					p	印刷	○
					c	コンバート	○
					s	書式作成	
013	eztab	表計算	Eztab	013	c	表組 コンバート	○
				212	c	コンバート	○
				013	e	編集	
					p	印刷	○
013	eztab	表計算	Eztab	212	g	グラフ作成	○
					c	コンバート	○

【図16】

71 ライブラリ ID	72 ライブラリ バイнда名	73 ライブラリ 名称	74 オープン 回数
020	/usr/lib/form	フォーム	0

【図17】

81 ユーザ ID	82 ユーザ 情報	83 ユーザのルーム ディレクトリ	84 ライブラリ 更新モード
1001	渡辺	/usr/users/watanabe	あり
1002	佐藤	/usr/users/sato	なし
1003	高田	/usr/users/takada	なし

【図18】

91 オフィス ID	92 ホスト名	93 オフィス名	94 ドメイン	95 ライブラリ 更新モード
10001	local	ローカル	68k	なし
10002	kaisya	会社	68k	なし

【図21】

141 ファイル名	142 ユーザID	143 アクセス レベル	144 日付
/tanaka/bunsho	1001	参照	91/04/02 15:55:24

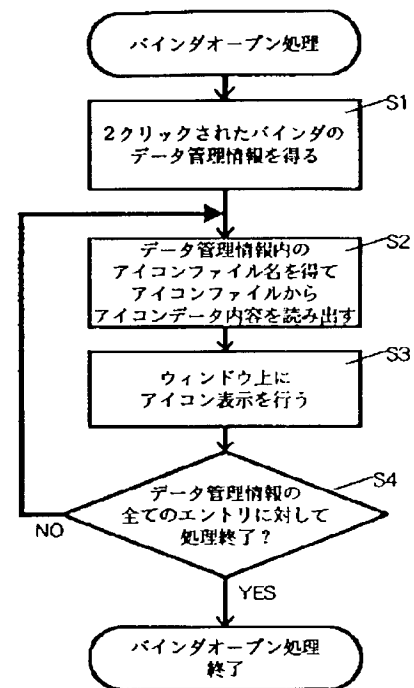
【図22】

150 検索順番1	検索順番2
ユーザID: 1001	オフィスID: 100001

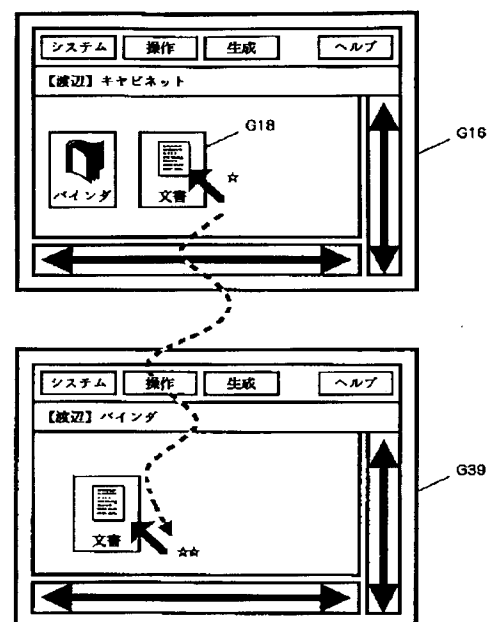
【図64】

161	ファイル名	cut_buffer
162	フォーマットID	012
163	サイズ	2012
164	所有者ユーザID	1001
165	所有者グループID	100001

【図24】



【図25】

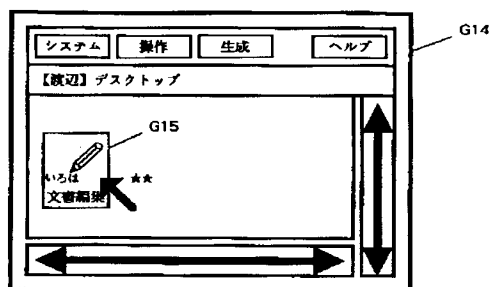


【図20】

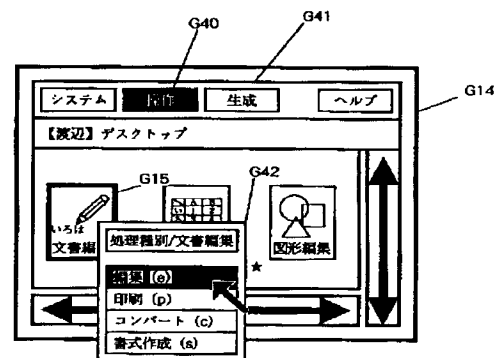
110

111	オブジェクト タイプ		データ	データ
112	ファイル名		bunsho.etx	form.etx
113	フォーマットID		012	012
114	クリエイタID		012	012
115	漢字名称		文書	フォーム
116	サイズ		2012	1208
117	登録日付		91/01/21 09:12:08	91/03/03 13:00:04
118	更新日付		91/03/30 11:52:50	91/03/03 13:00:04
119	参照日付		91/04/02 15:55:24	91/04/02 16:01:18
120	所有者ユーザID		1001	1001
121	所有者グループID		100001	100001
122	参照ユーザID		1001	1002
123	更新ユーザID		1001	1001
124	アクセスレベル	参照	所有者	OK
グループ			OK	
その他			OK	
更新		所有者	OK	
		グループ	OK	
		その他	OK	
125	ドメイン		68k	68k
126	公開先ユーザID		なし	なし
127	公開先グループID		なし	なし
128	公開複写制限		0	なし
129	公開期限		91/04/30 00:00:00	なし
130	データ廃棄期限		91/04/30 23:59:59	なし
131	タグ参照先		なし	なし

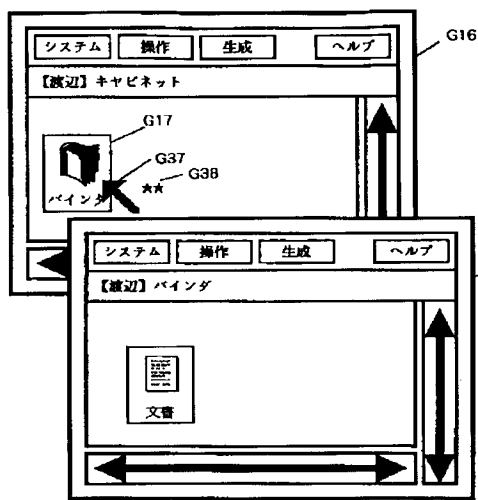
【図27】



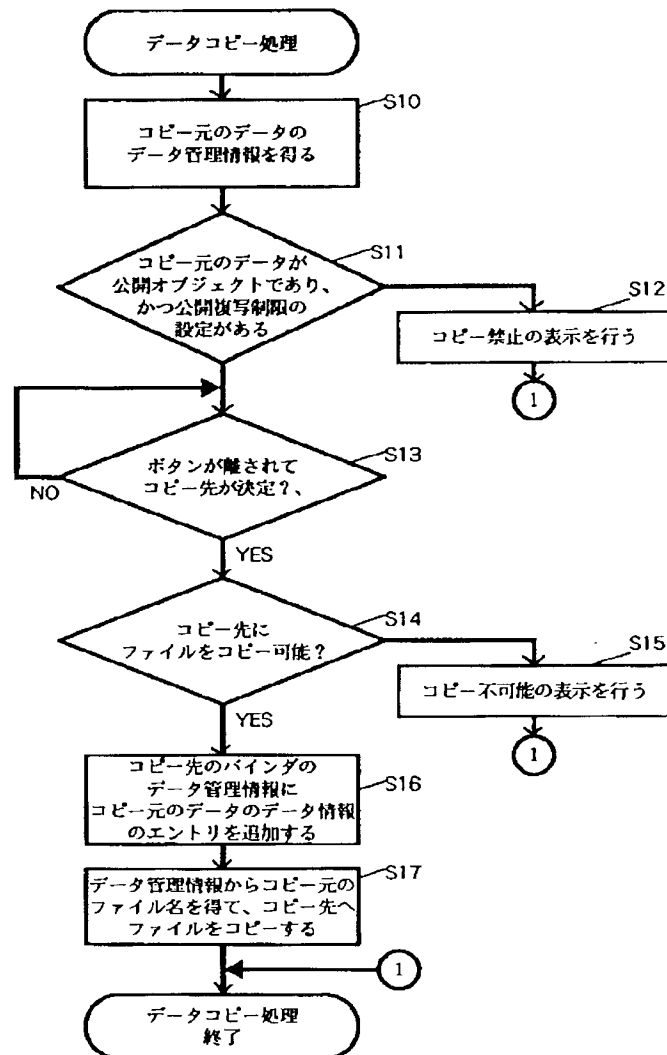
【図29】



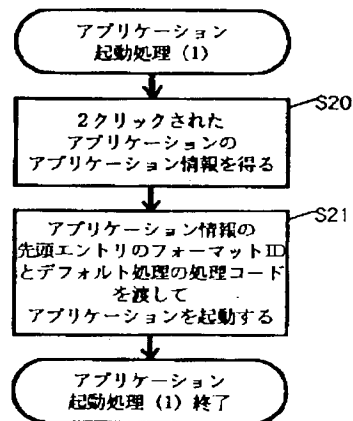
【図23】



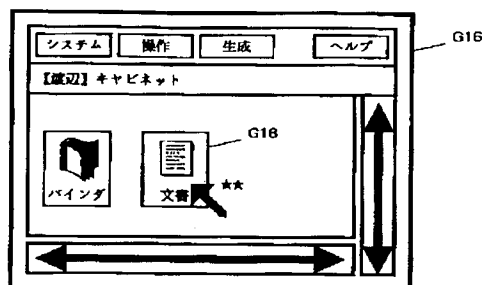
【図26】



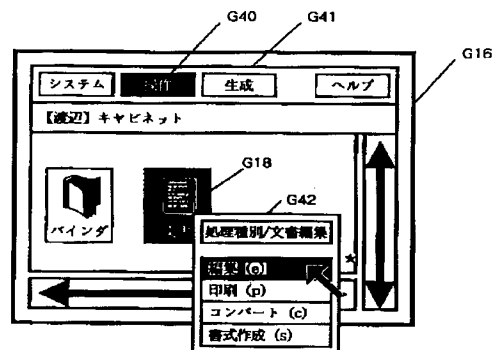
【図28】



【図32】

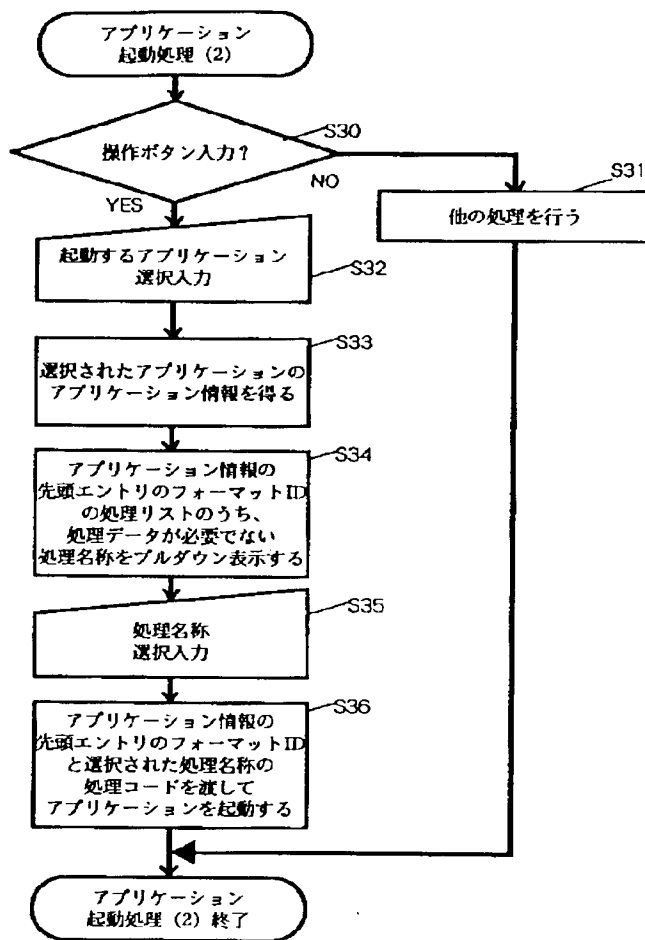


【図34】

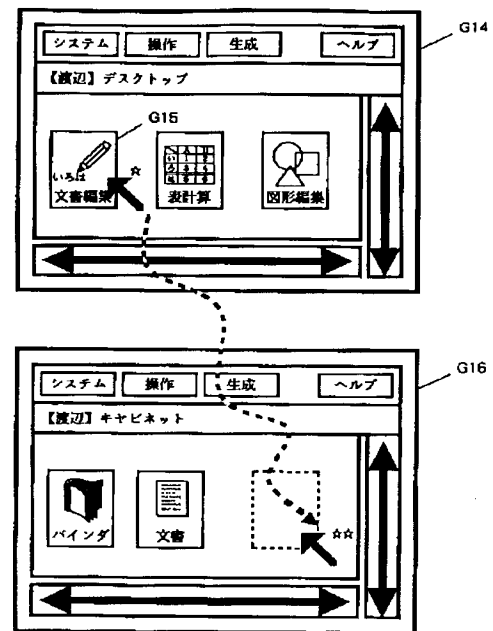




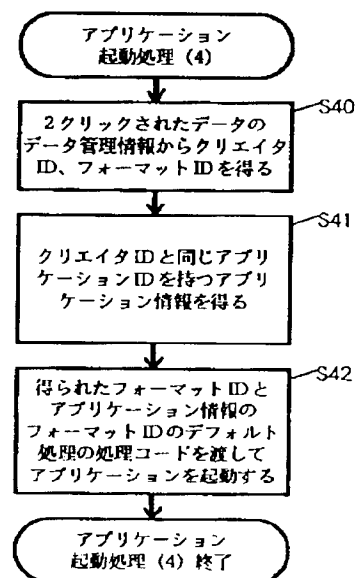
【図30】



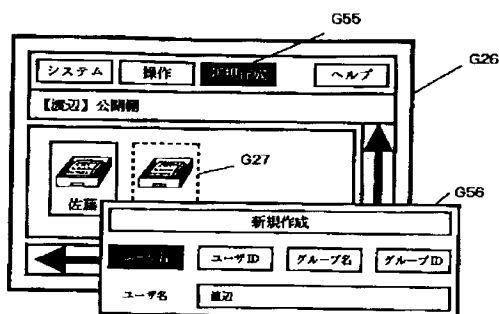
【図31】



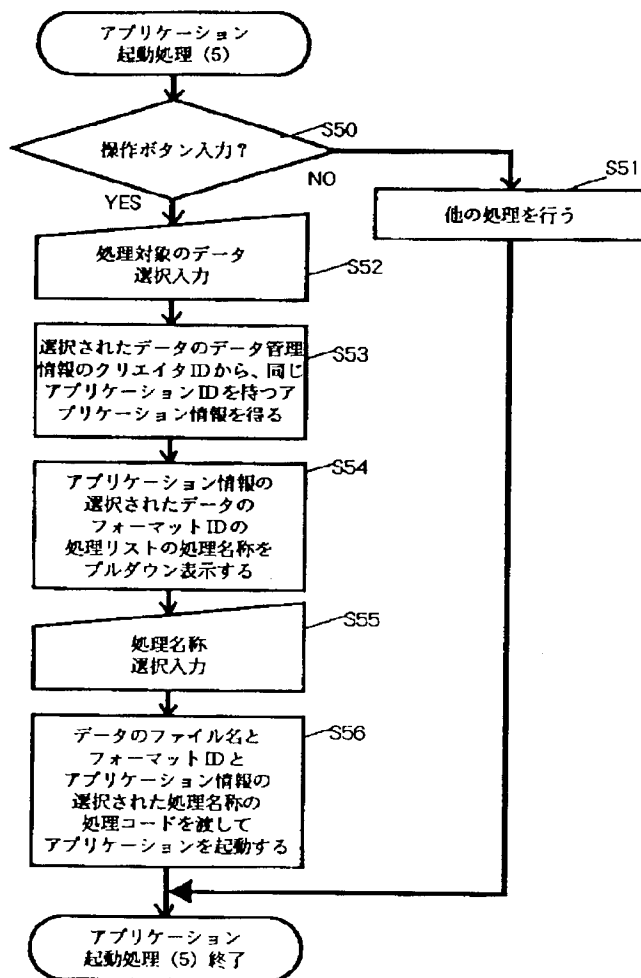
【図33】



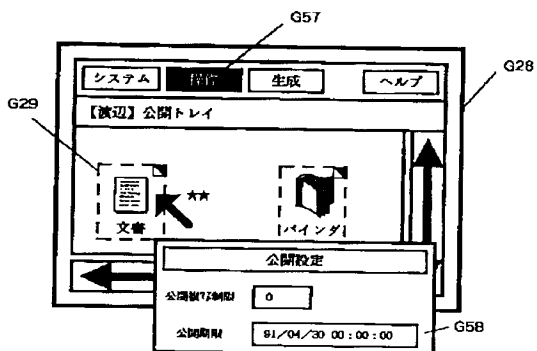
【図46】



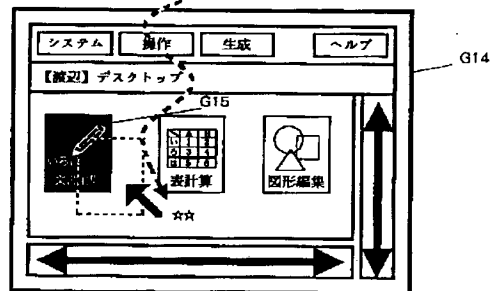
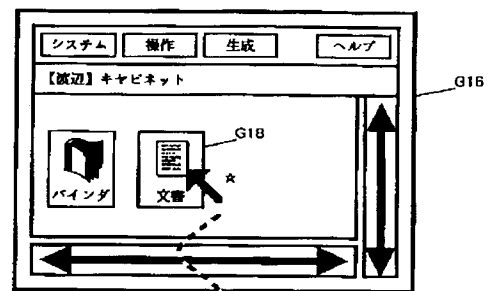
【図35】



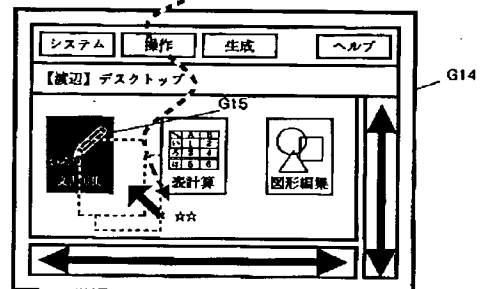
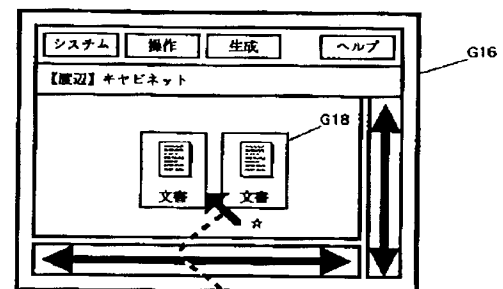
【図47】



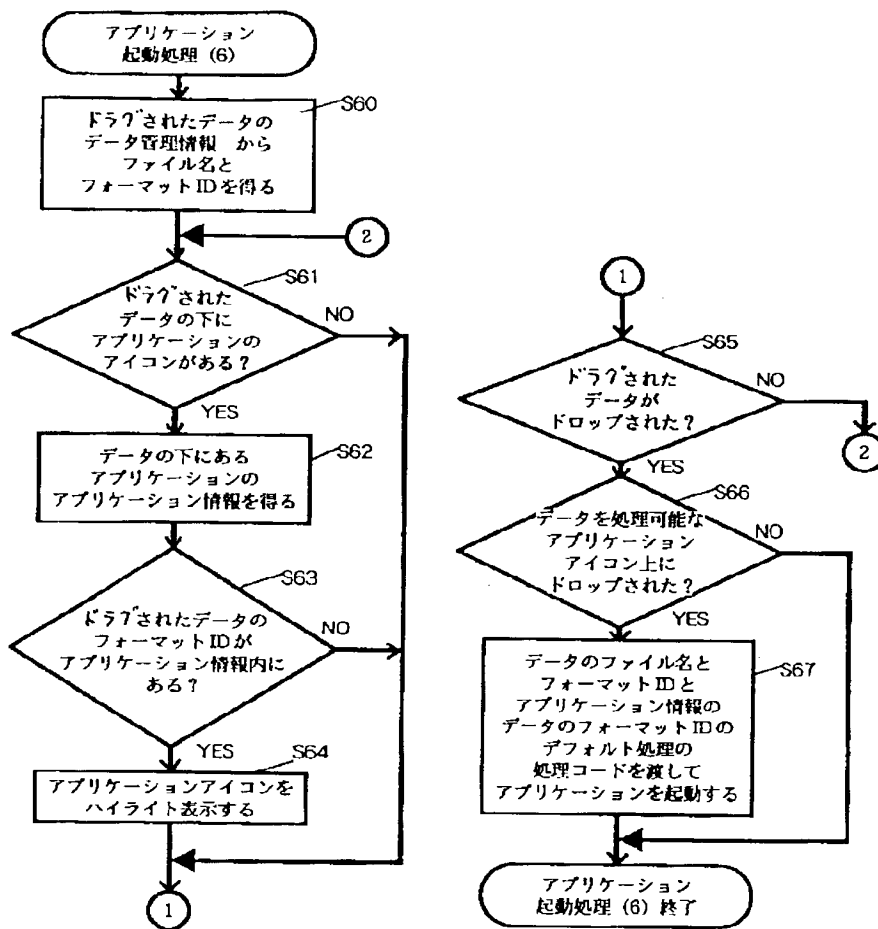
【図36】



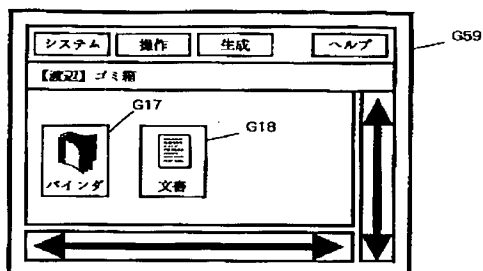
【図38】



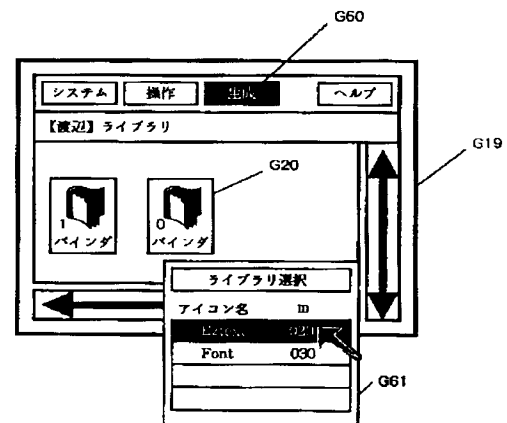
【図37】



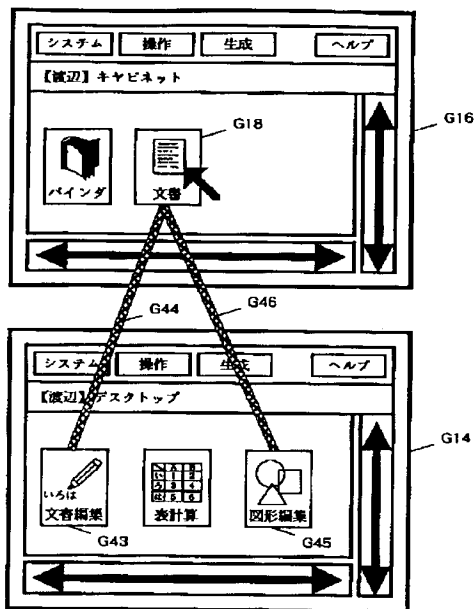
【図54】



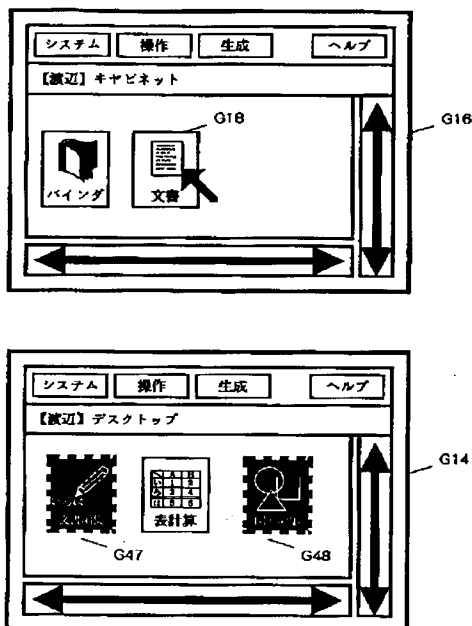
【図57】



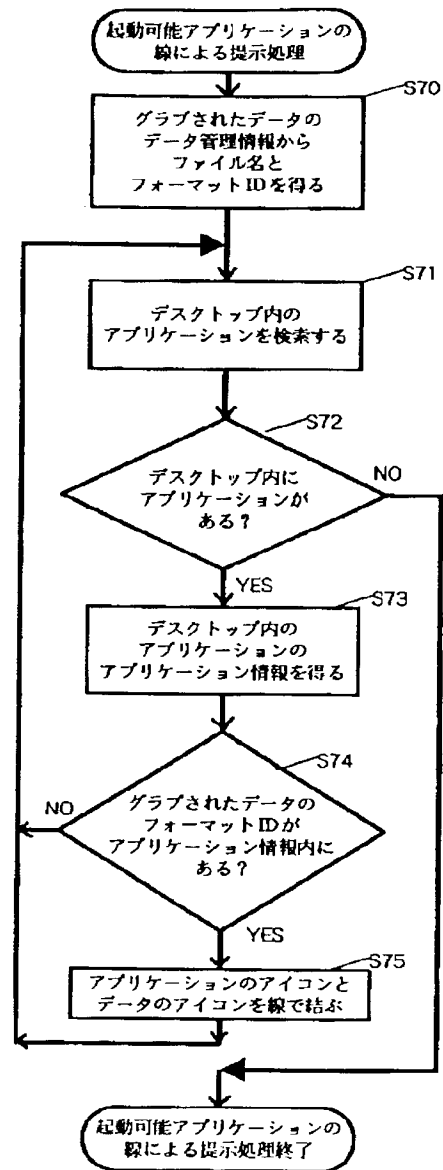
【図39】



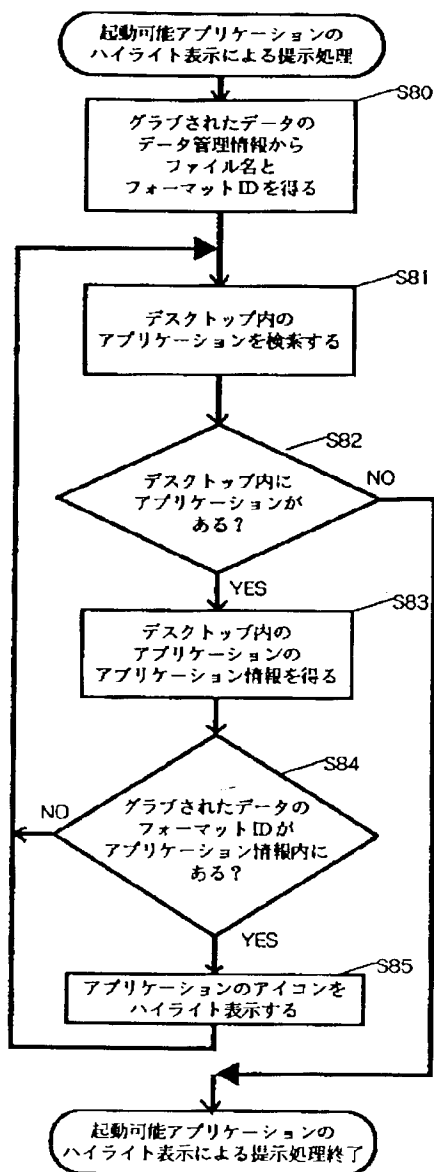
【図41】



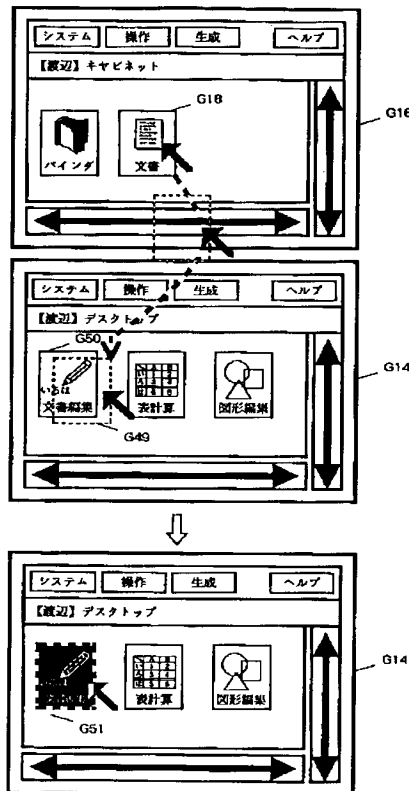
【図40】



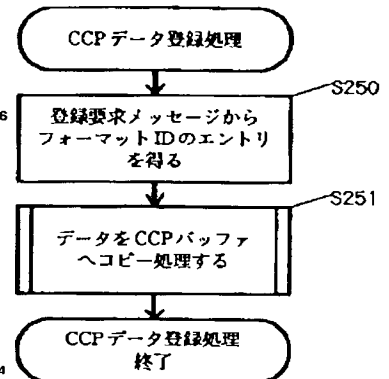
【図42】



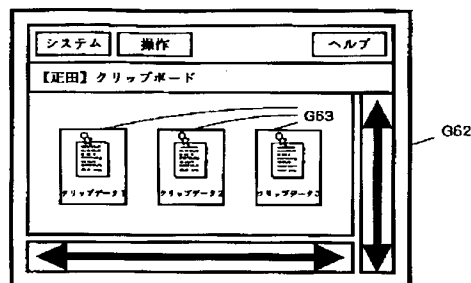
【図43】



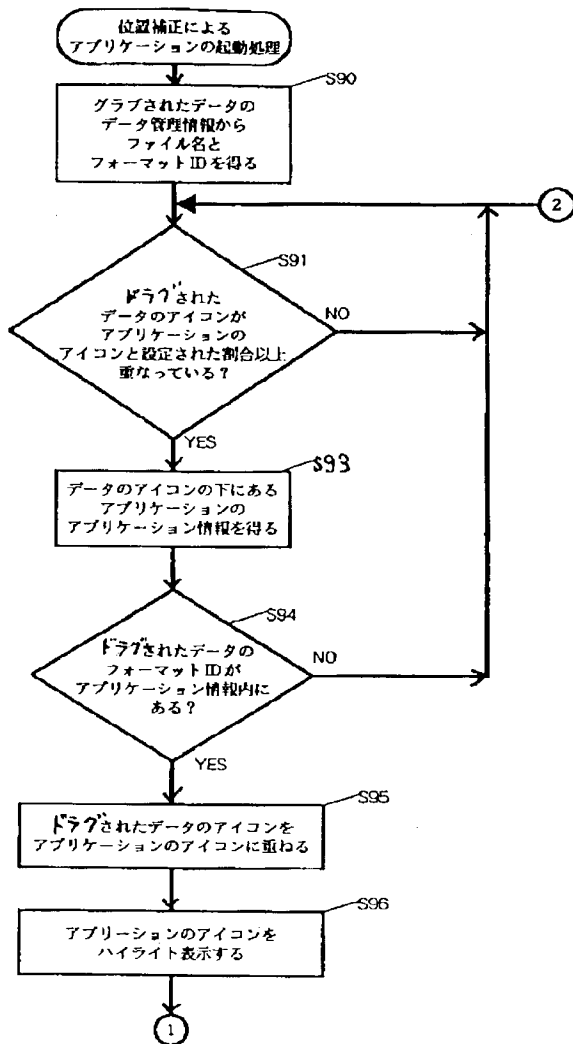
【図65】



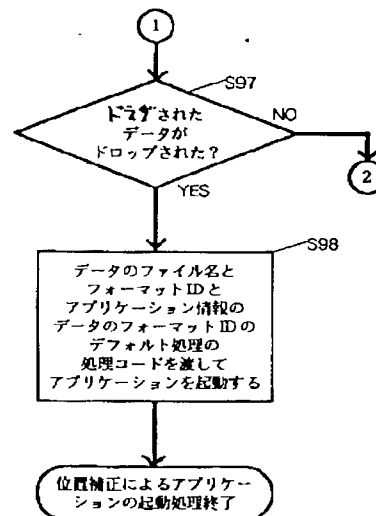
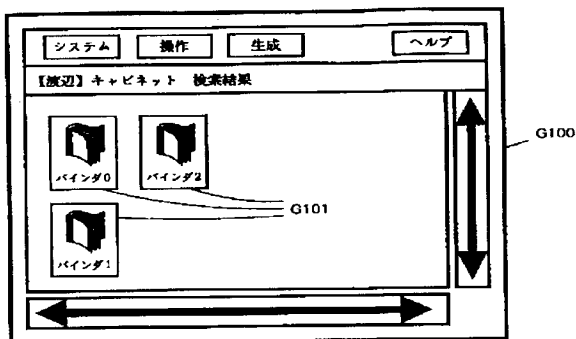
【図61】



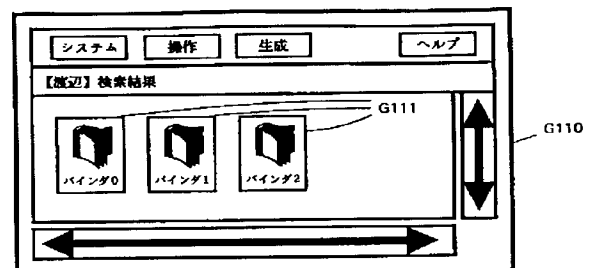
【図44】



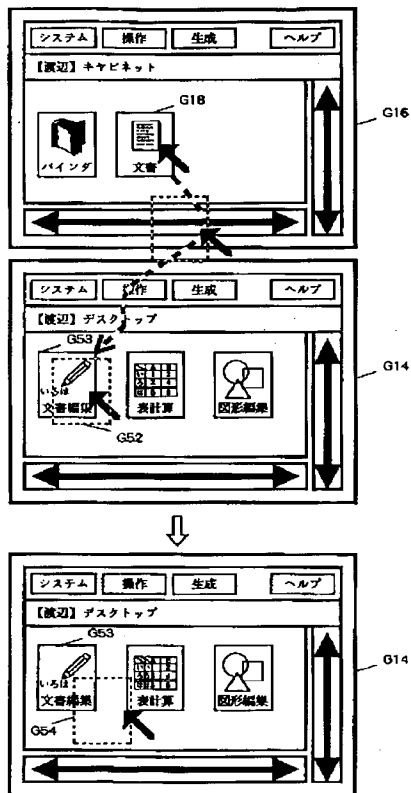
【図67】



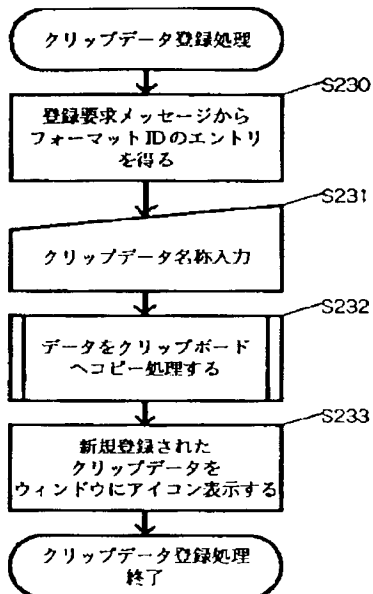
【図69】



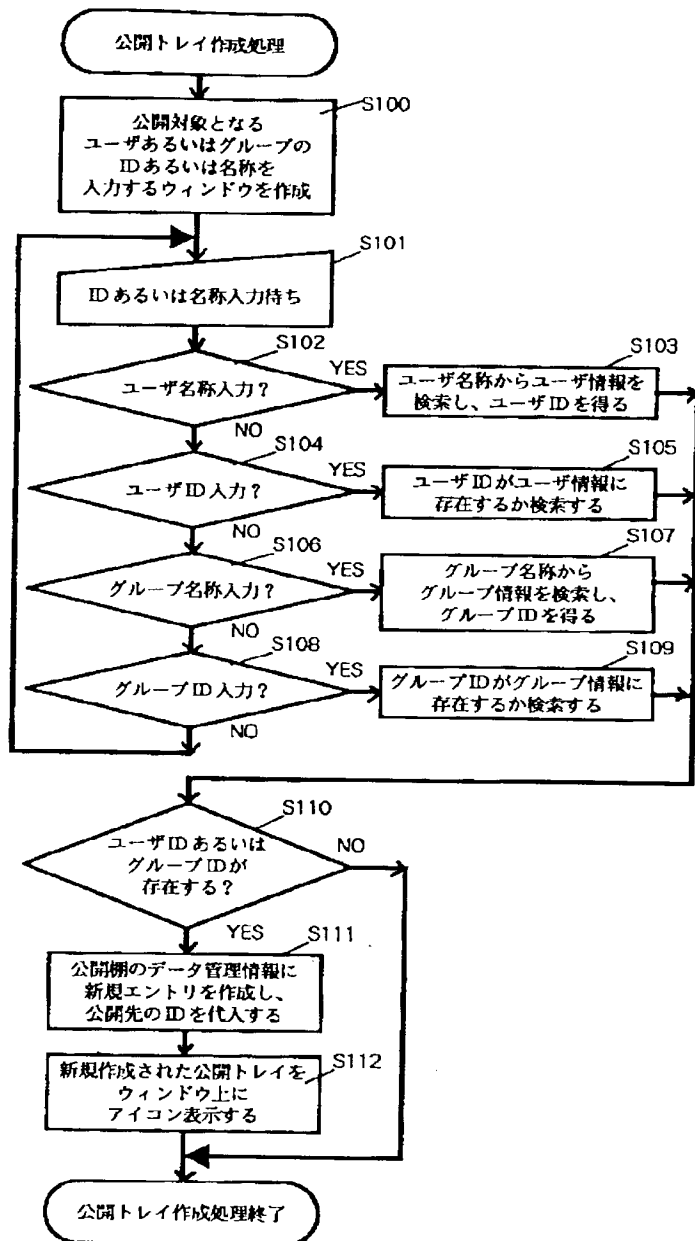
【図45】



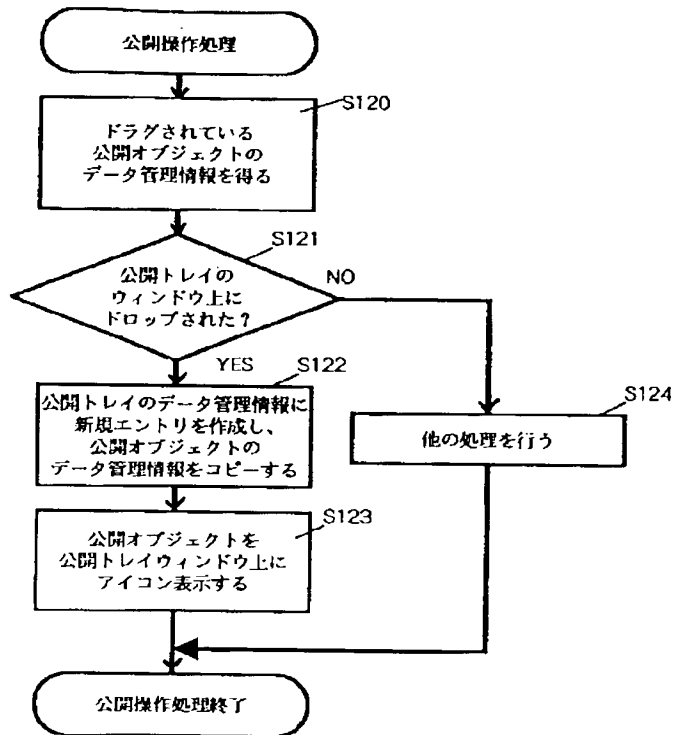
【図62】



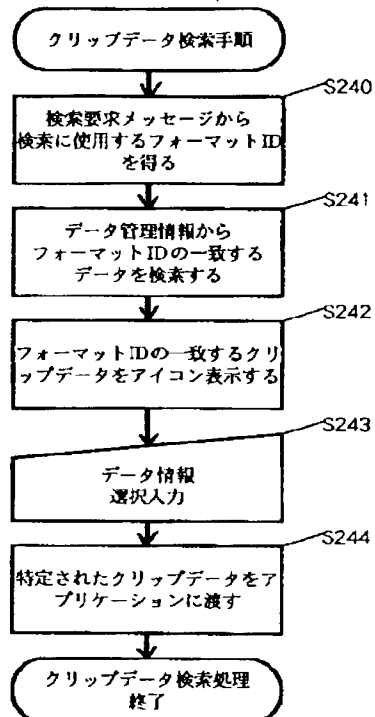
【図48】



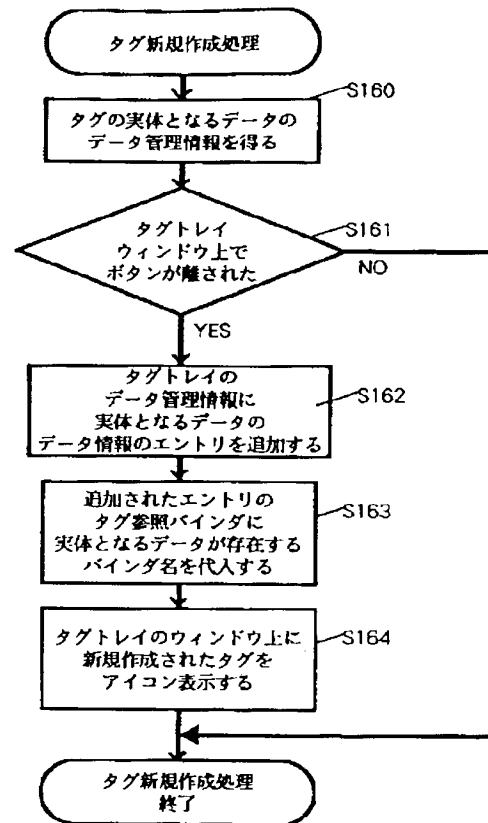
【図49】



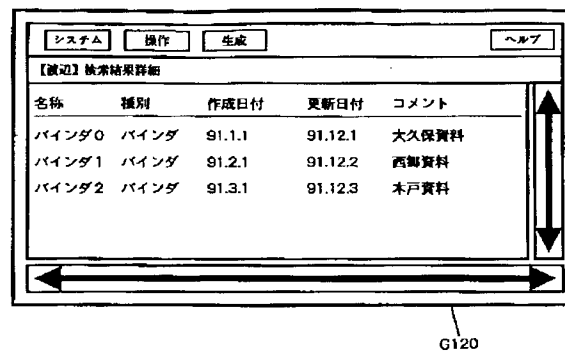
【図63】



【図52】

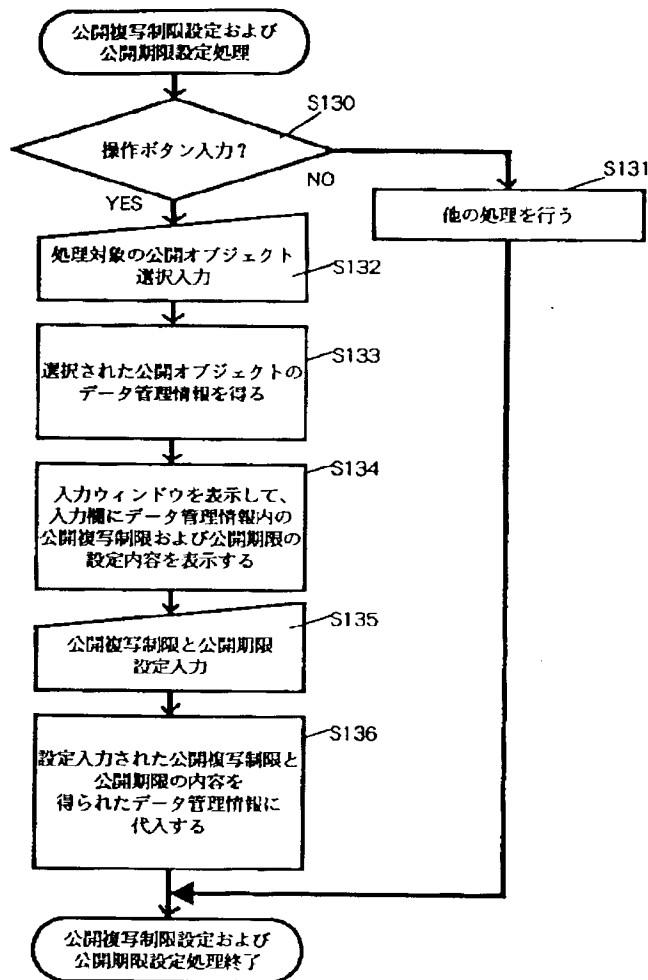


【図71】

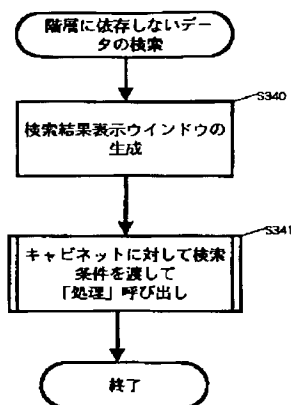




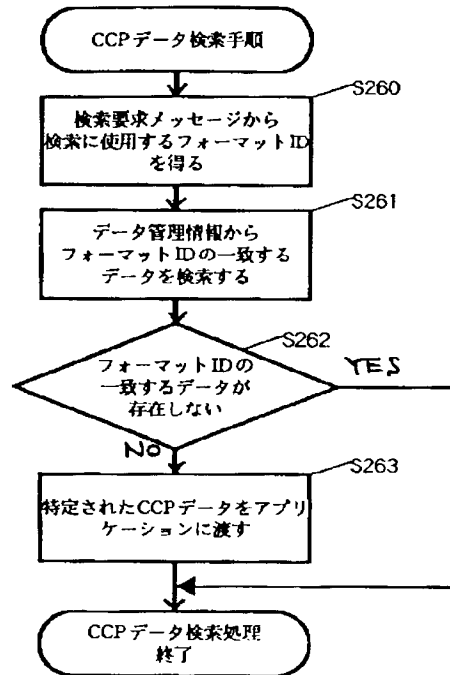
【図50】



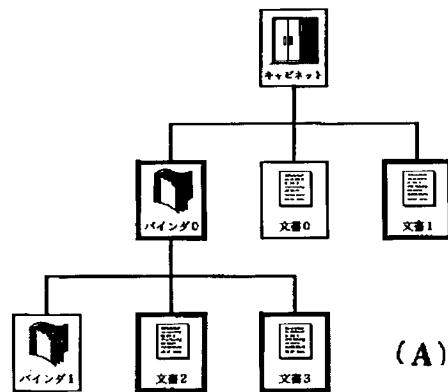
【図74】



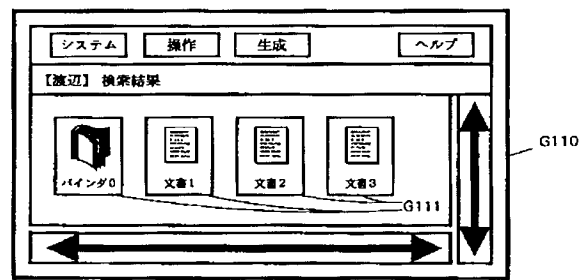
【図66】



【図79】

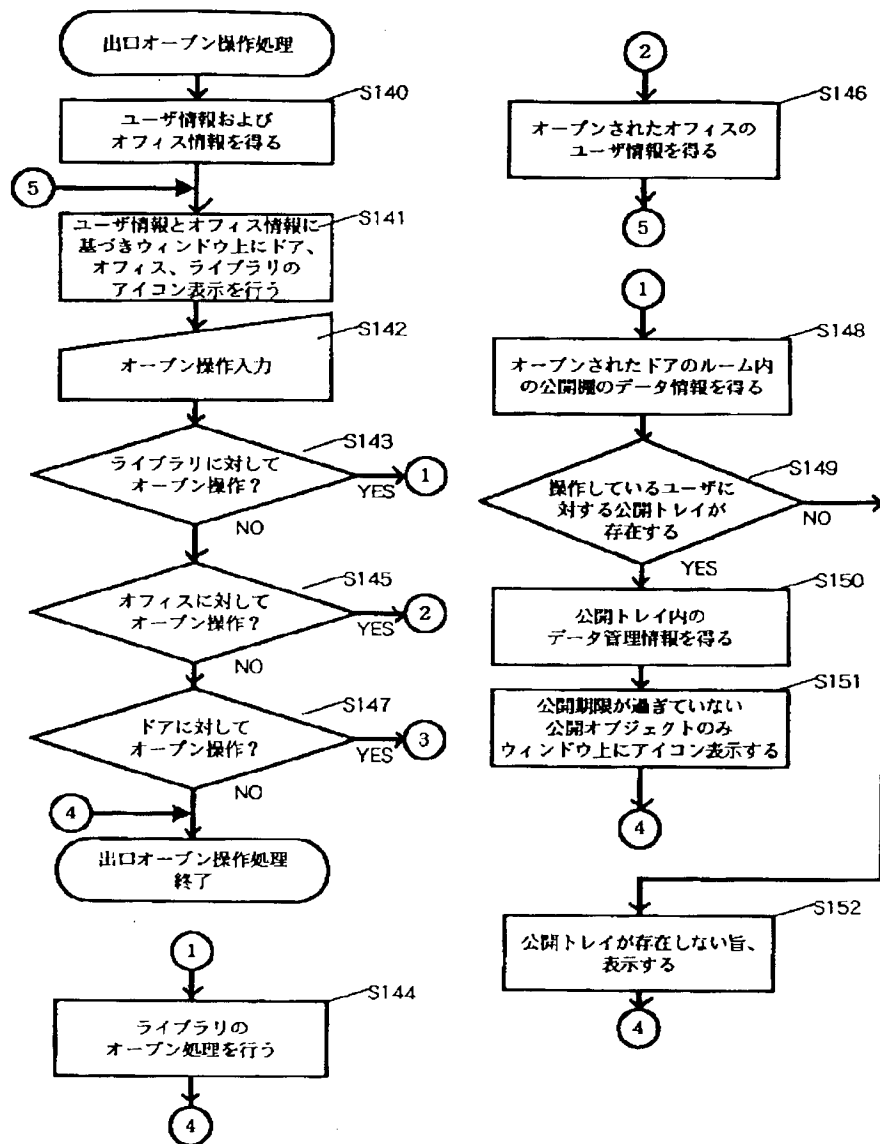


(A)

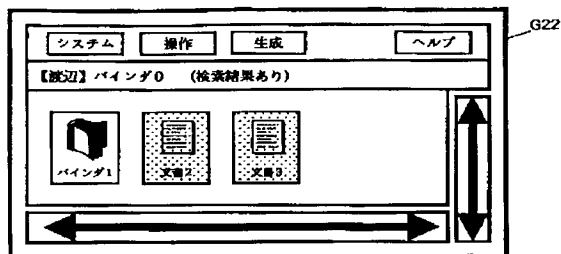


(B)

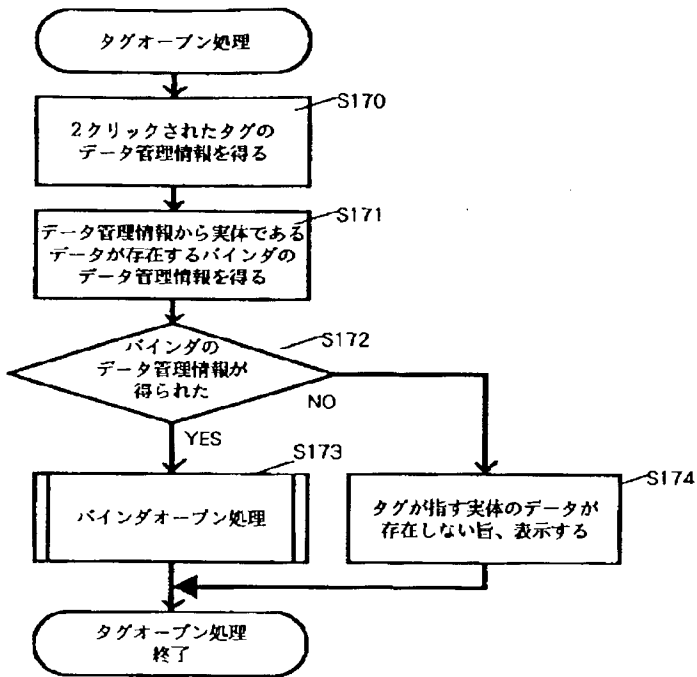
【図51】



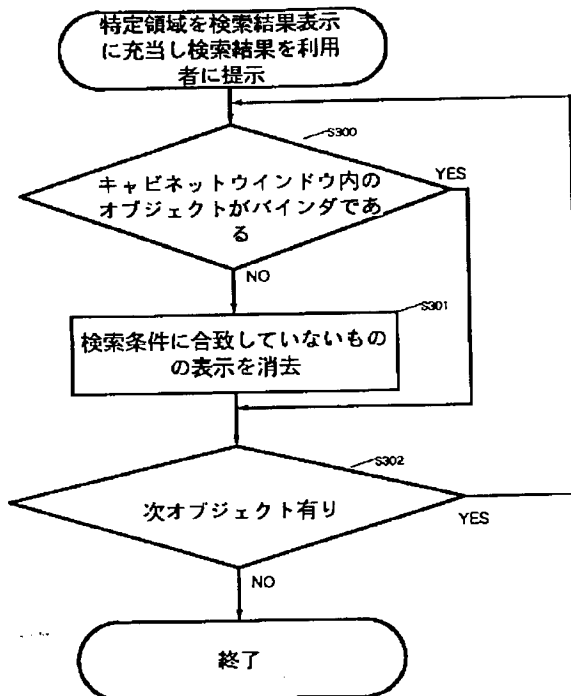
【図80】



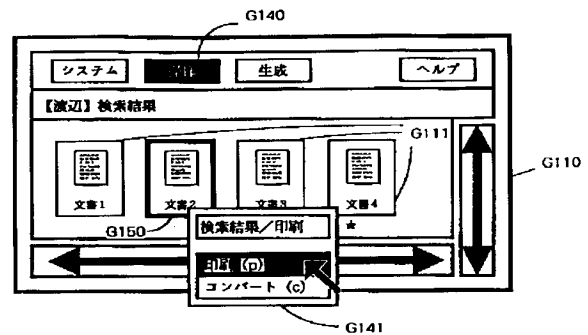
【図53】



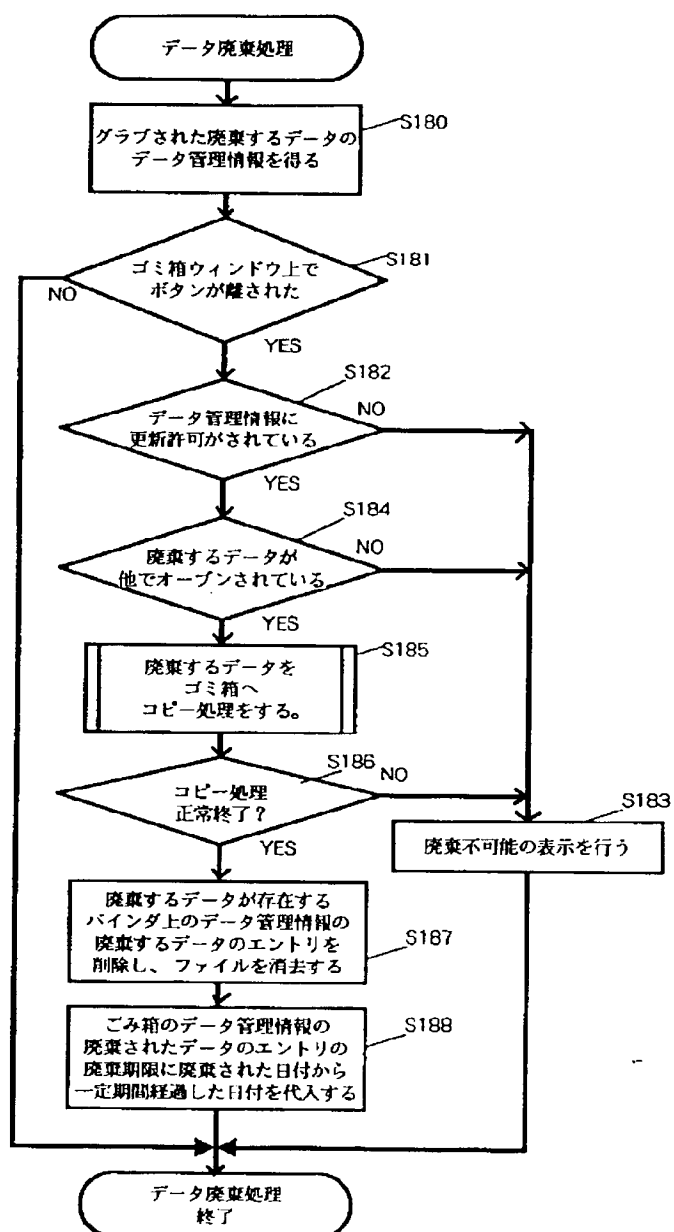
【図68】



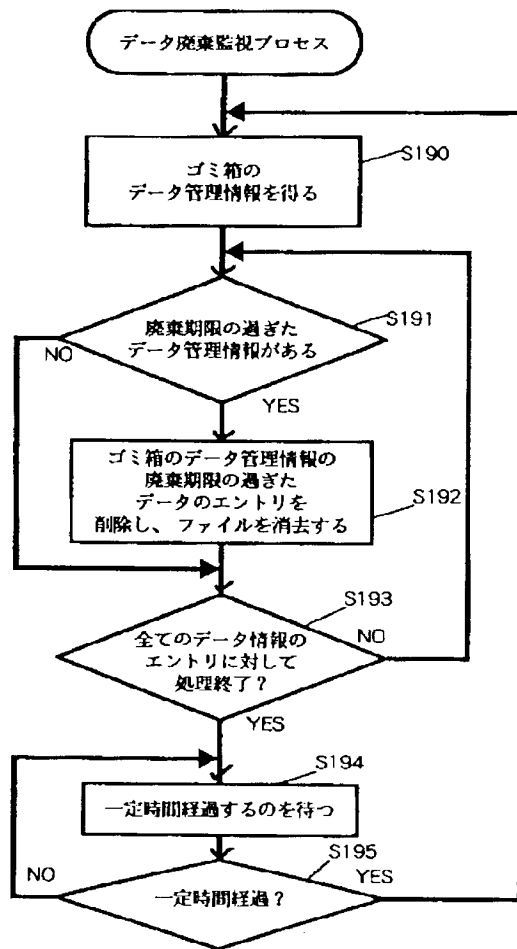
【図82】



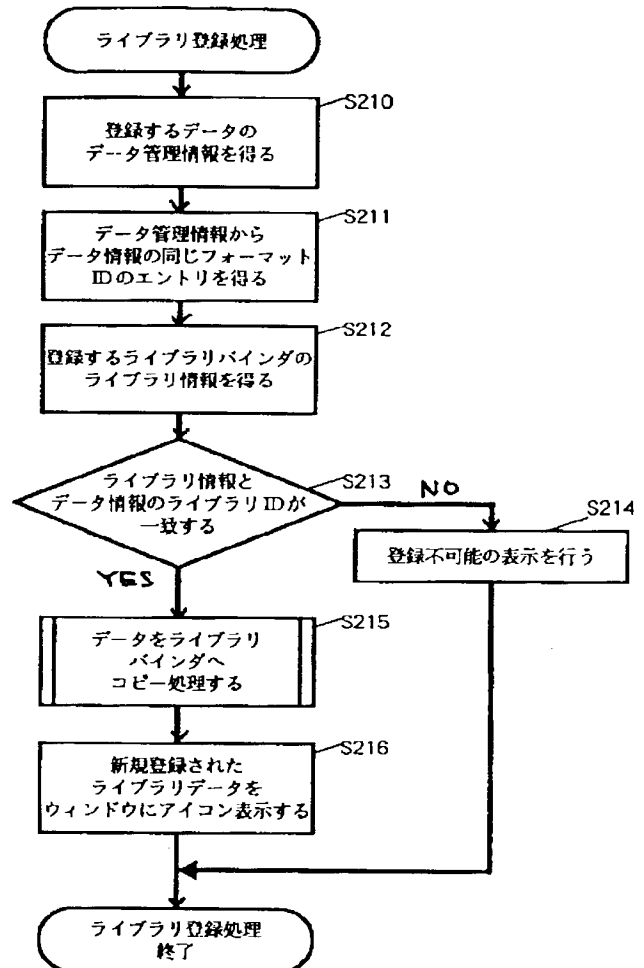
【図55】



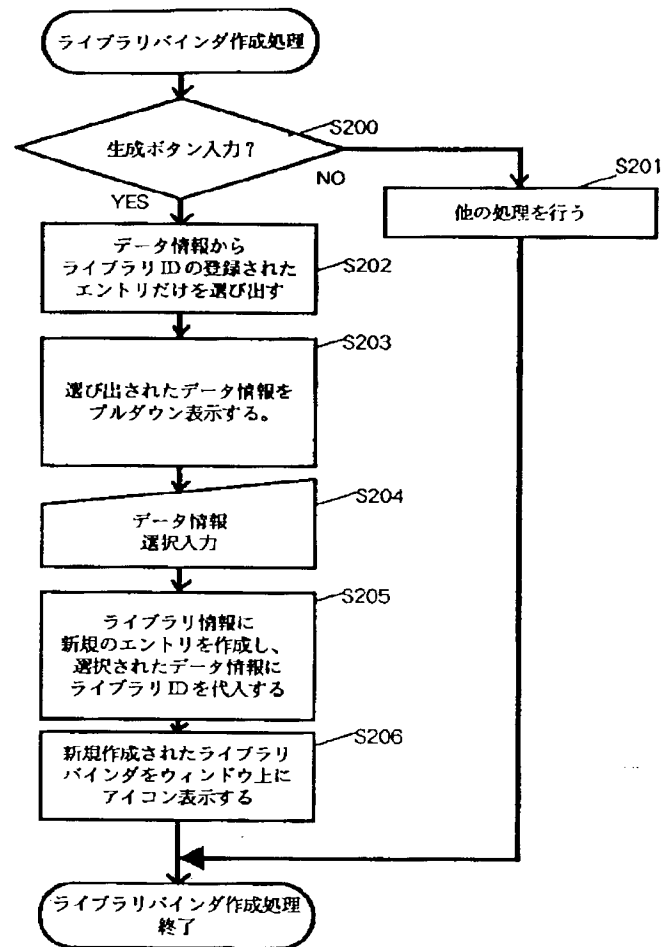
【図56】



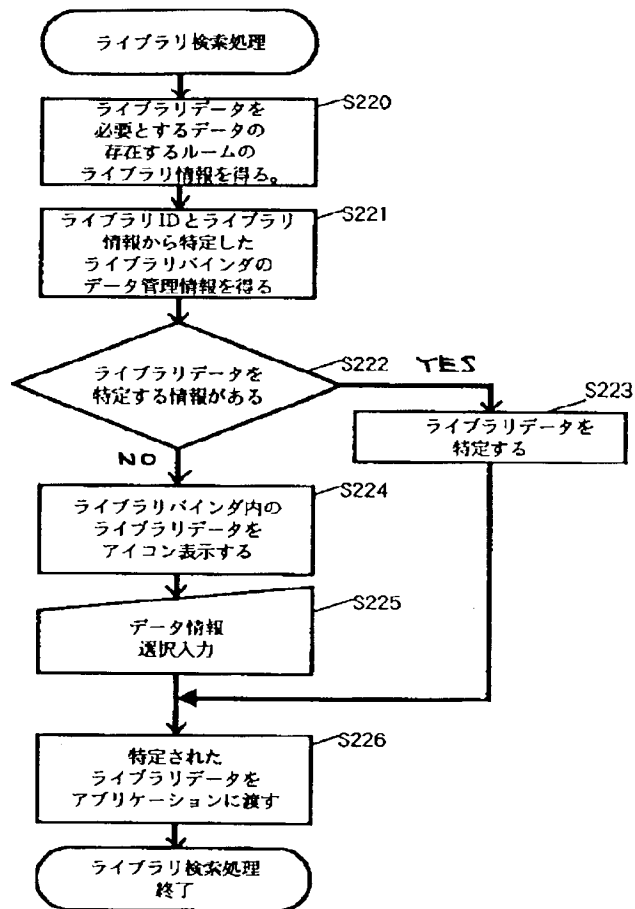
【図59】



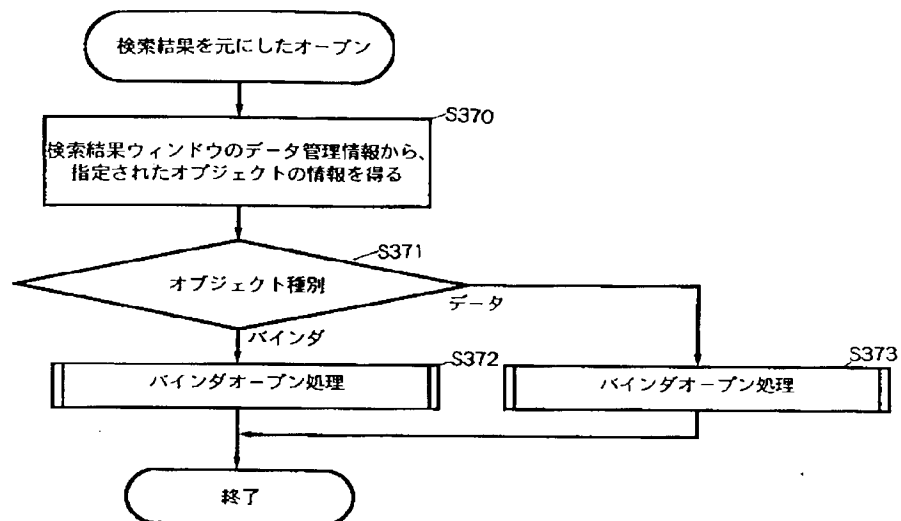
【図58】



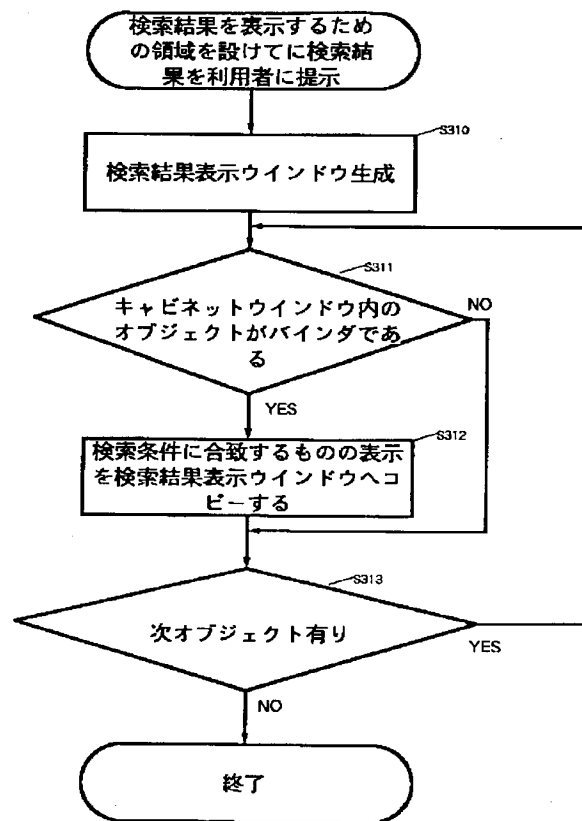
【図60】



【図77】

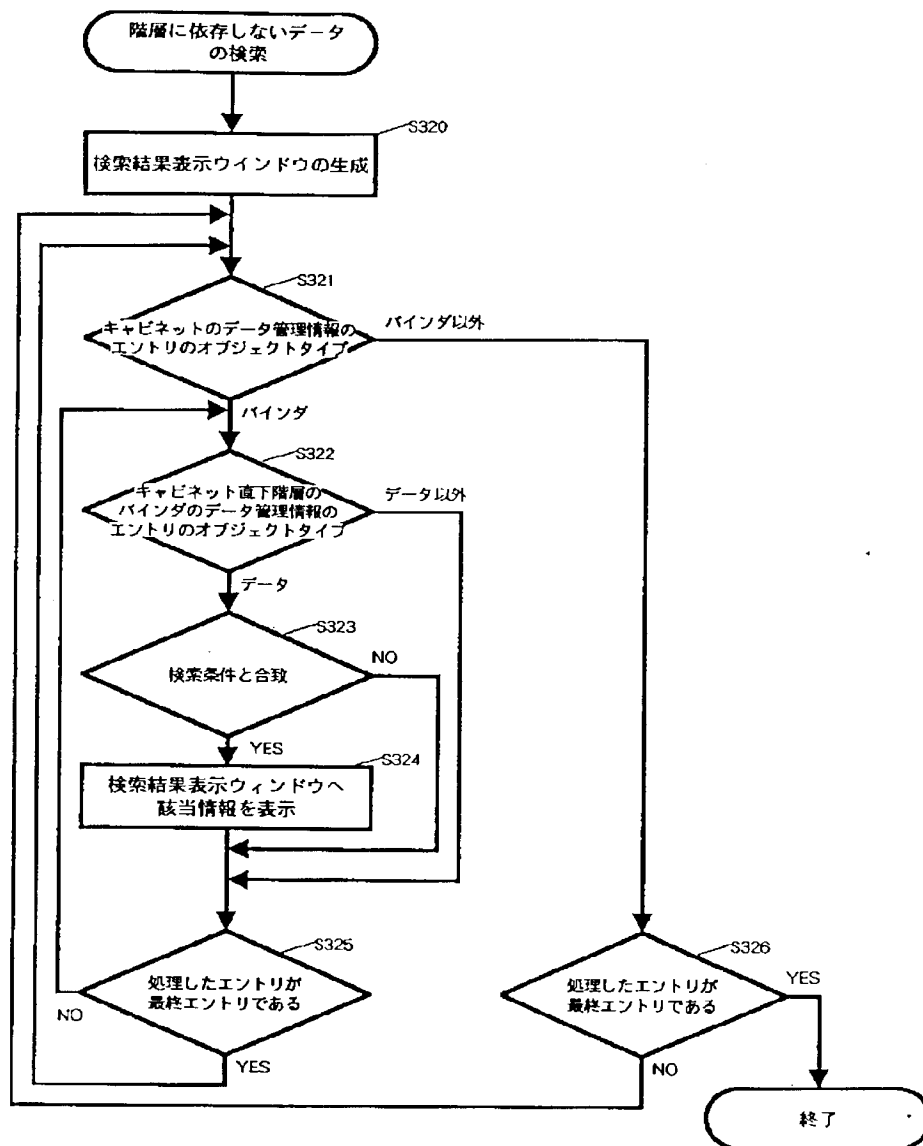


【図70】

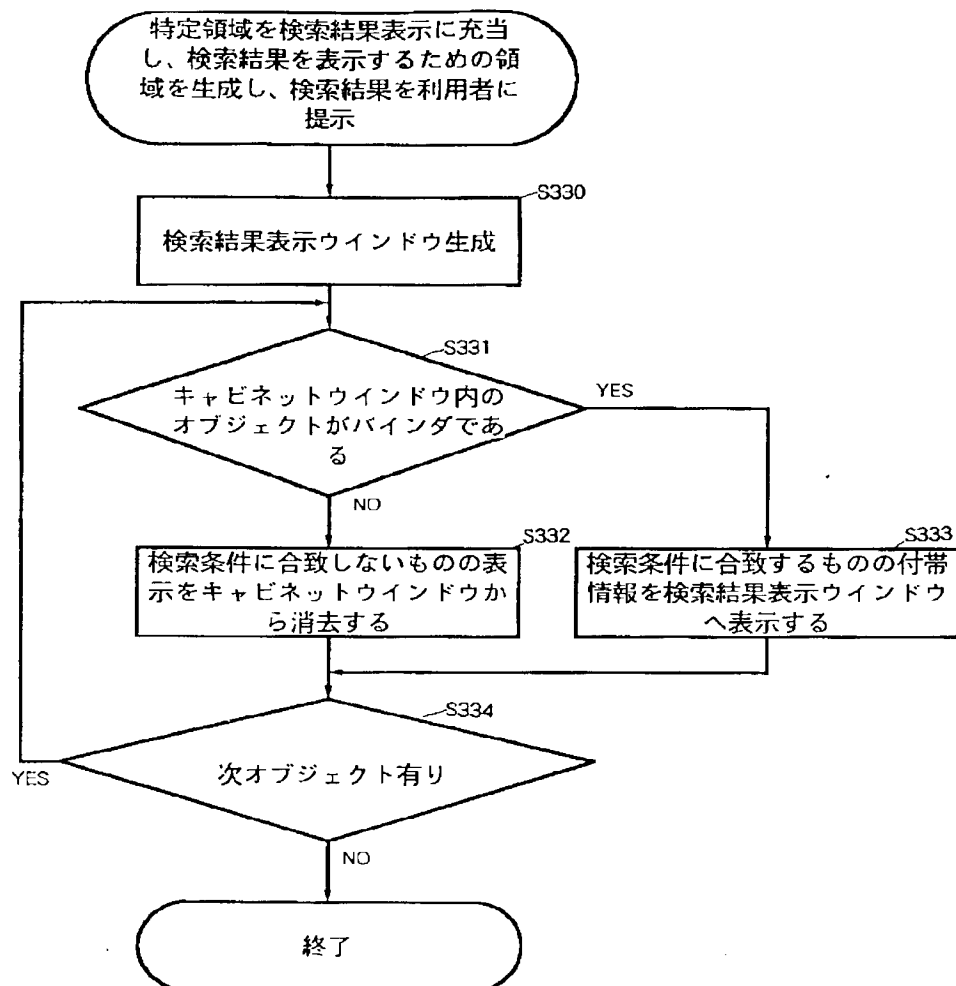




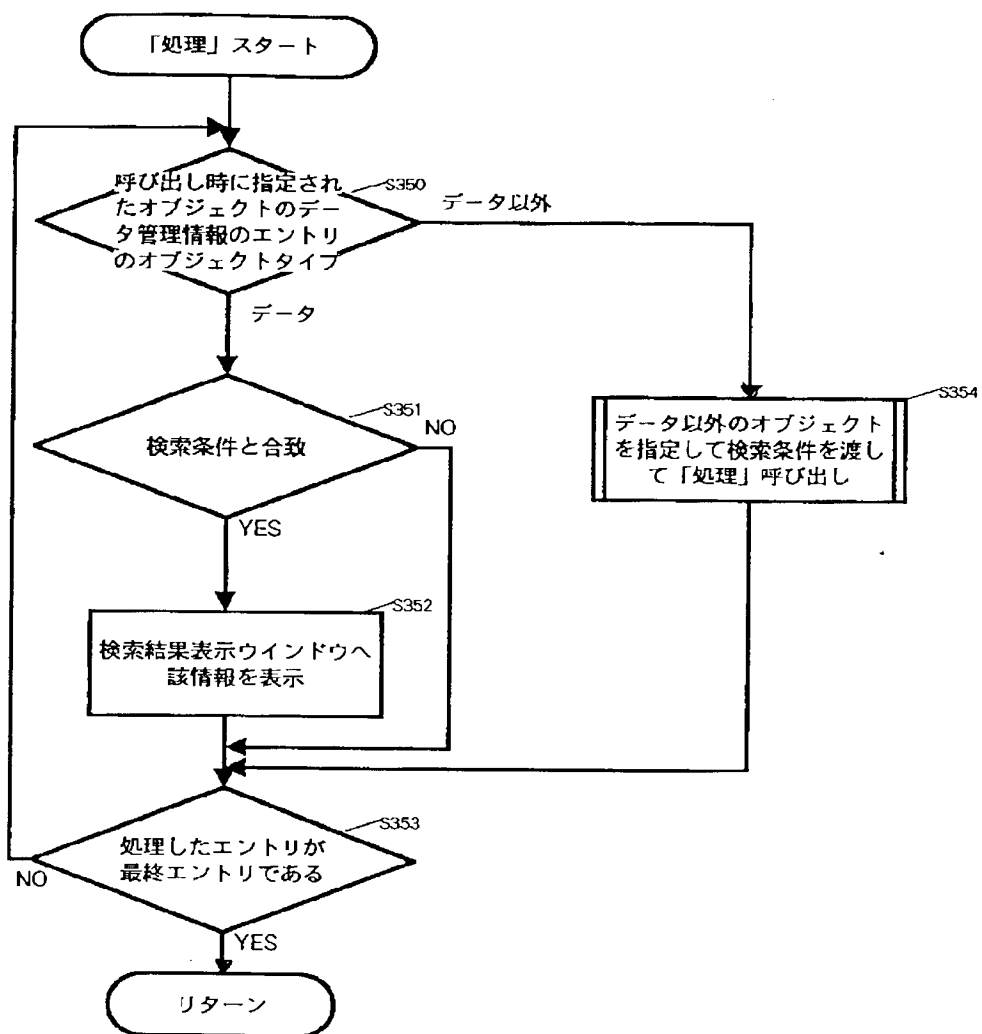
【図72】



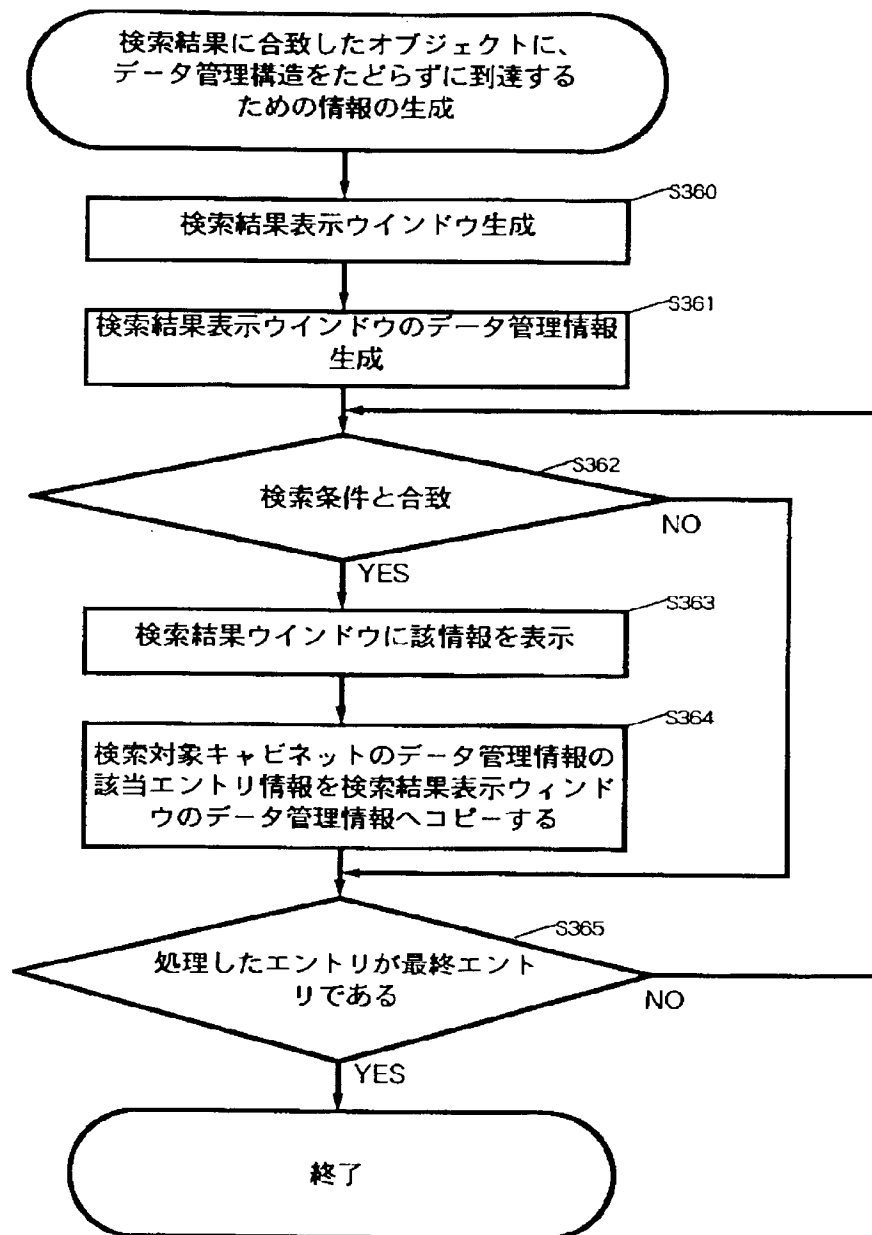
【図73】



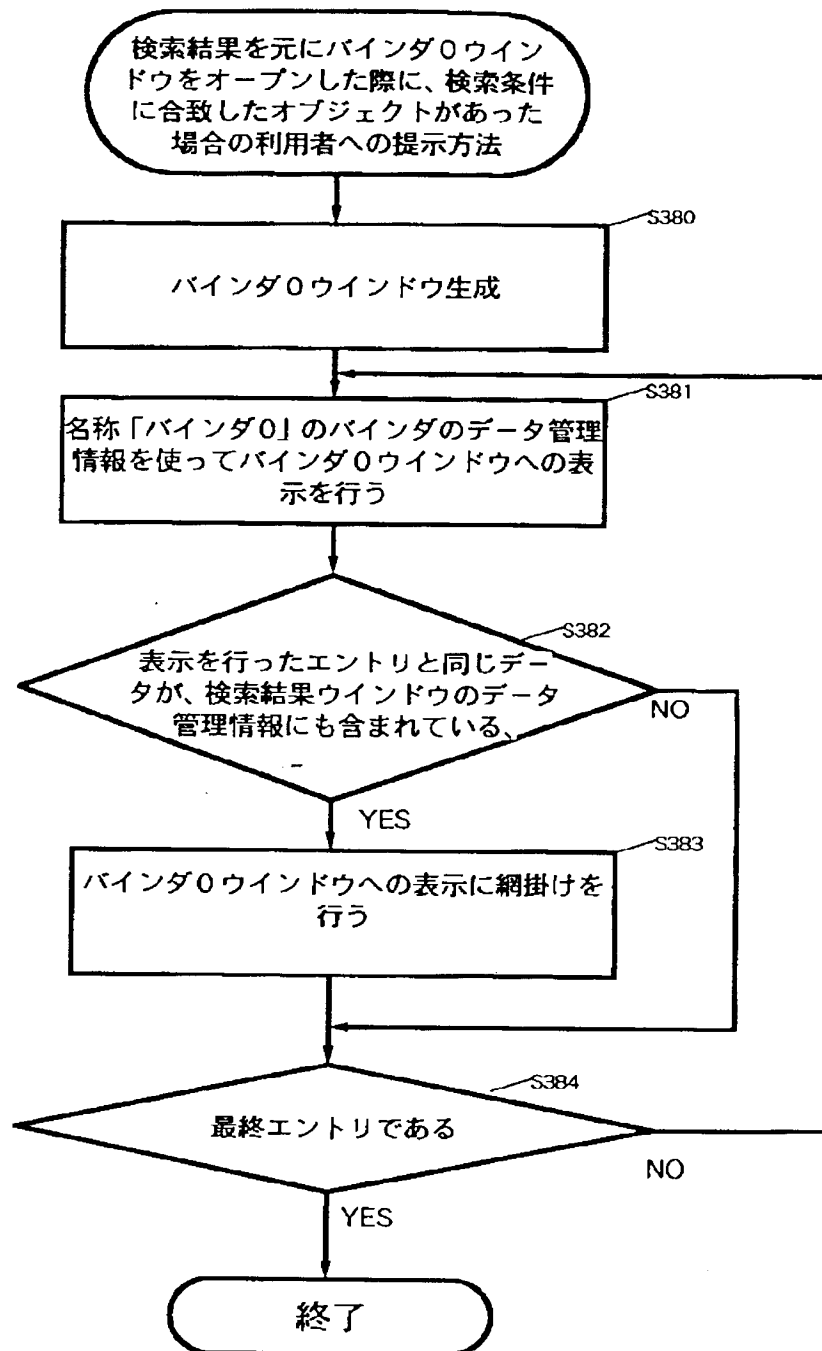
【図75】



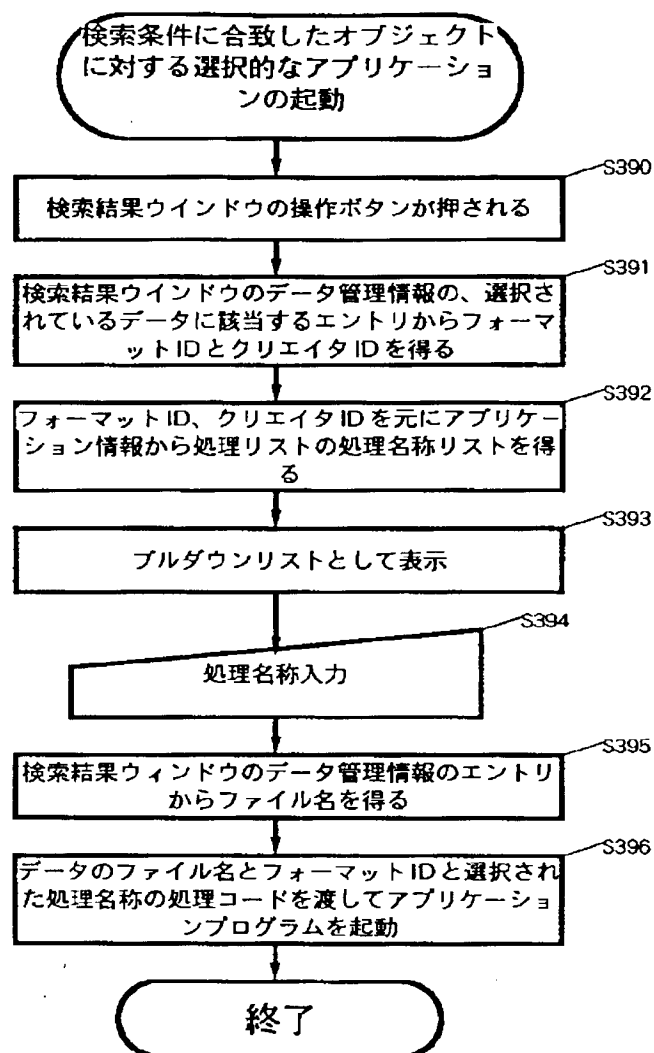
【図76】



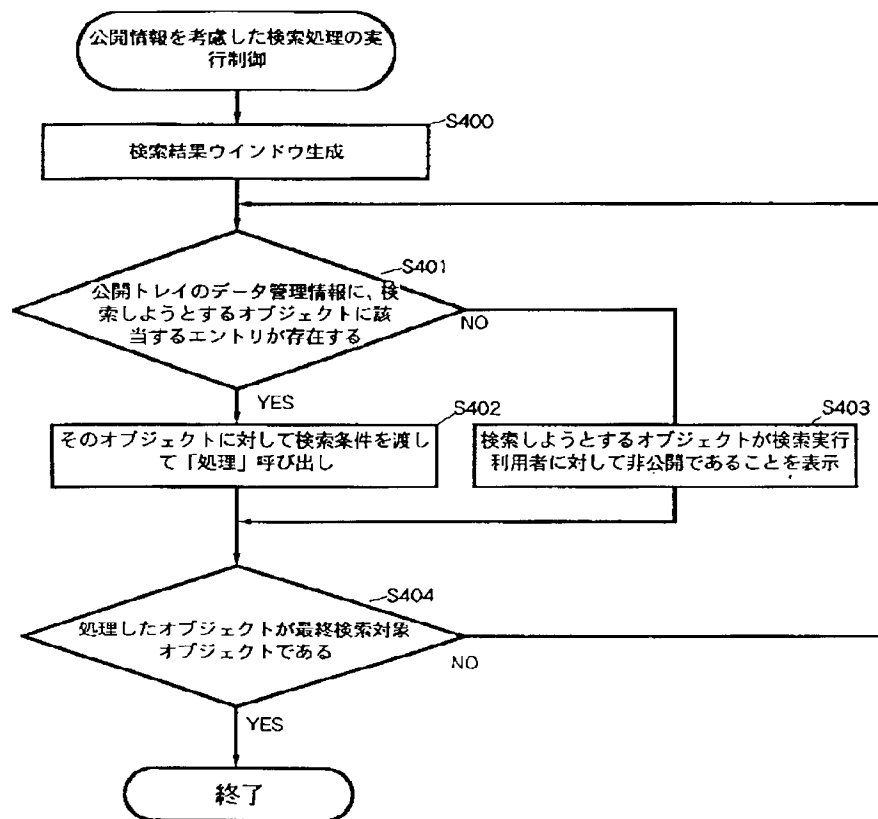
【図78】



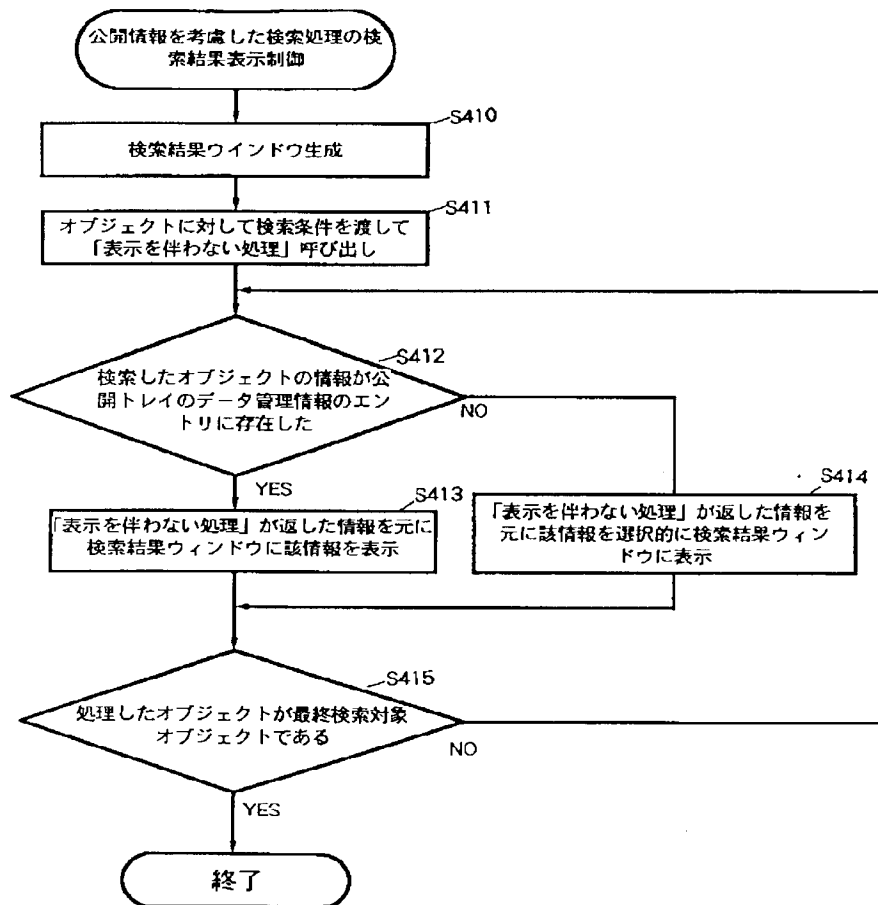
【図81】



【図83】

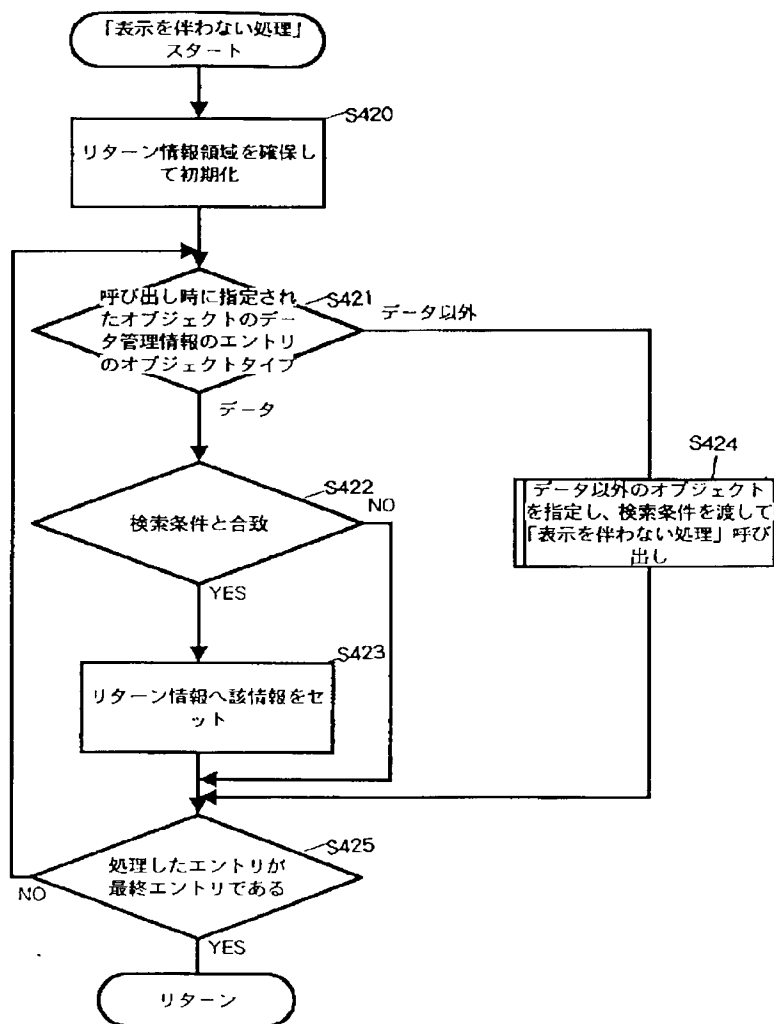


【図84】

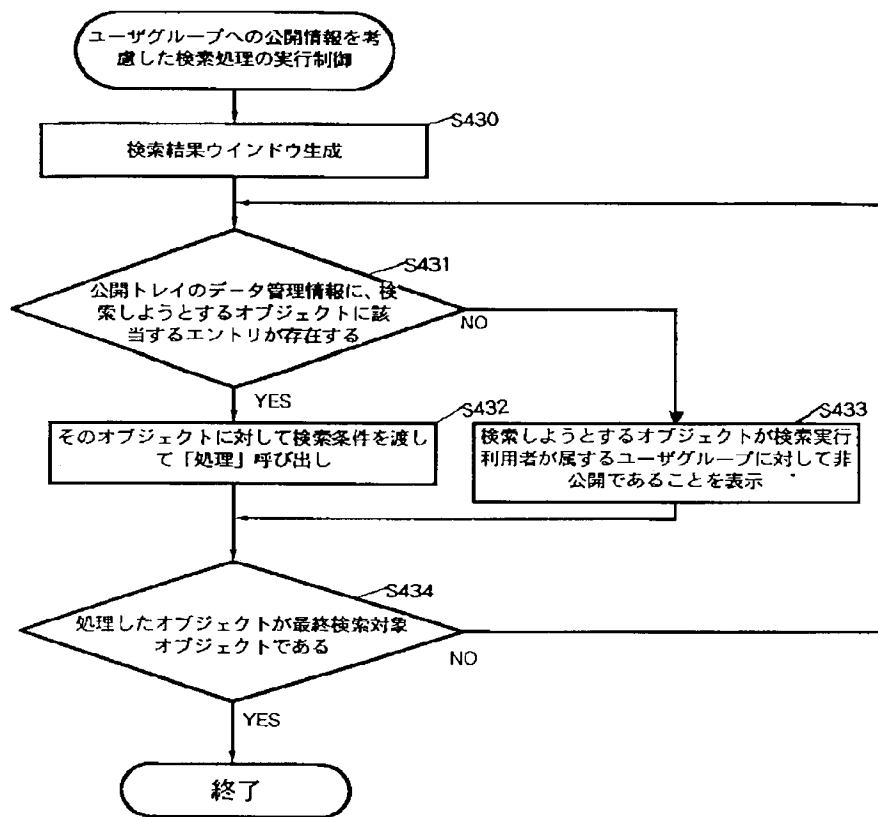




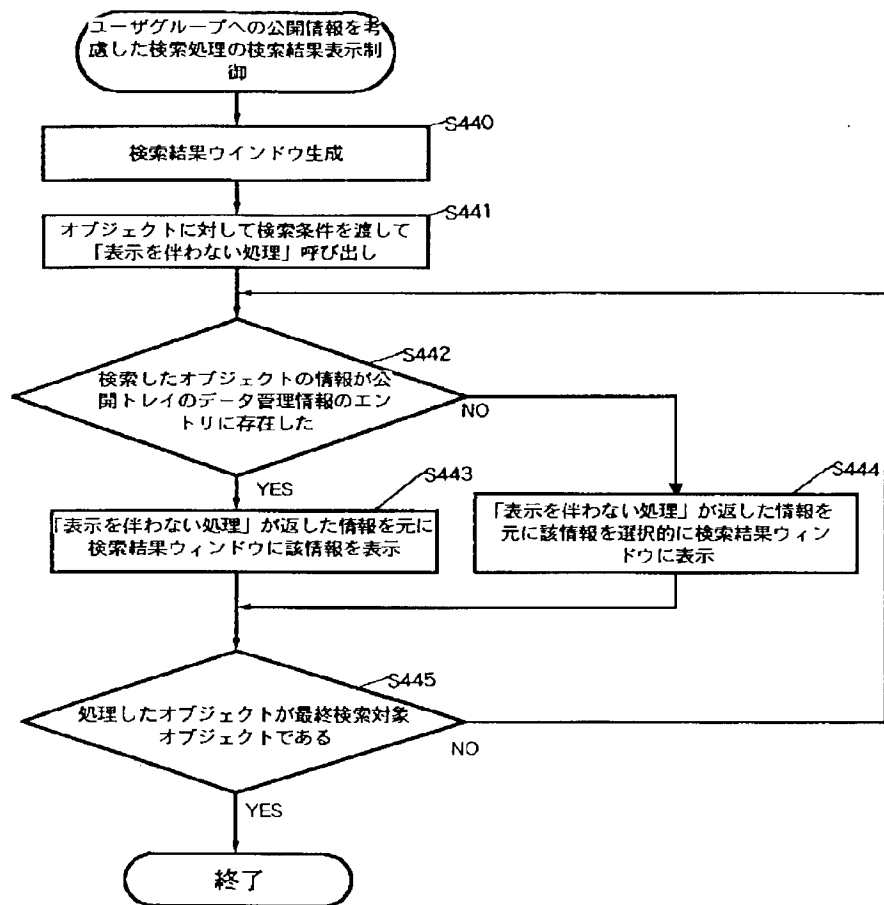
【図85】



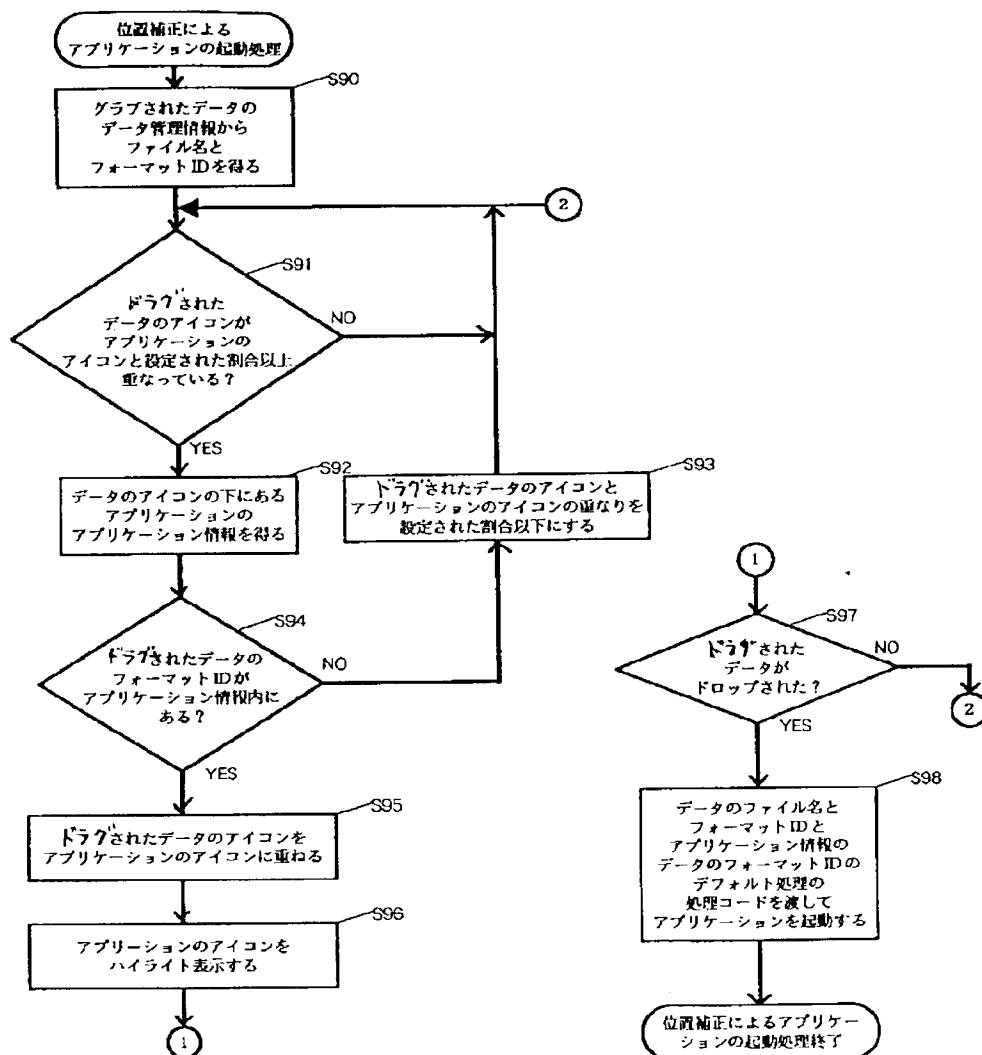
【図86】



【図87】



【図88】



フロントページの続き

(72) 発明者 高田 康文  
 東京都大田区下丸子3丁目30番2号 キヤ  
 ノン株式会社内

【公報種別】特許法第17条の2の規定による補正の掲載  
【部門区分】第6部門第3区分  
【発行日】平成13年1月19日（2001. 1. 19）

【公開番号】特開平6-131145  
【公開日】平成6年5月13日（1994. 5. 13）  
【年通号数】公開特許公報6-1312  
【出願番号】特願平4-131137  
【国際特許分類第7版】

G06F	3/14	370
	12/00	515
	15/00	390

【F I】

G06F	3/14	370 A
	12/00	515 B
	15/00	390

【手続補正書】

【提出日】平成11年5月24日（1999. 5. 24）

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】発明の名称

【補正方法】変更

【補正内容】

【発明の名称】 データ管理方法及び装置

【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正内容】

【特許請求の範囲】

【請求項1】 アプリケーションプログラム等により作成されたデータをネットワーク上に接続された端末間で管理するデータ管理装置であって、前記データを記憶する記憶手段と、前記記憶手段に記憶されたデータから所望のデータを指示する指示手段と、前記指示手段により指示されたデータを複数の端末に公開するよう制御する制御手段とを備えることを特徴とするデータ管理装置。

【請求項2】 特定の端末を指定する指定手段を更に備え、前記制御手段は、前記指示手段により指示されたデータを前記指定手段により指定された特定の端末に公開するよう制御することを特徴とする請求項1に記載のデータ管理装置。

【請求項3】 前記制御手段は、前記データを前記指定手段により指定された端末から参照或は更新を許可することを特徴とする請求項2に記載のデータ管理装置。

【請求項4】 前記データに対する公開期限を設定する設定手段を更に備えることを特徴とする請求項2に記載のデータ管理装置。

【請求項5】 前記公開されたデータに対して、複写を制限することを特徴とする請求項4に記載のデータ管理装置。

【請求項6】 前記指定手段により指定された端末に公開するデータを、特定領域に表示することを特徴とする請求項2に記載のデータ管理装置。

【請求項7】 前記特定領域にデータを移動すること、で、該データを前記指定手段により指定された端末に公開することを特徴とする請求項2に記載のデータ管理装置。

【請求項8】 アプリケーションプログラム等により作成されたデータをネットワーク上に接続された端末間で管理するデータ管理方法であって、前記データを記憶手段に記憶する記憶工程と、前記記憶手段に記憶されたデータから所望のデータを指示する指示工程と、前記指示工程により指示されたデータを複数の端末に公開するよう制御する制御工程とを備えることを特徴とするデータ管理方法。

【請求項9】 特定の端末を指定する指定工程を更に備え、前記制御工程は、前記指示工程により指示されたデータを前記指定工程により指定された特定の端末に公開するよう制御することを特徴とする請求項8に記載のデータ管理方法。

【請求項10】 前記制御工程は、前記データを前記指定工程により指定された端末から参照或は更新を許可することを特徴とする請求項9に記載のデータ管理方法。

【請求項11】 前記データに対する公開期限を設定す

る設定工程を更に備えることを特徴とする請求項9に記載のデータ管理方法。

【請求項12】 前記公開されたデータに対して、複写を制限することを特徴とする請求項11に記載のデータ管理方法。

【請求項13】 前記指定工程により指定された端末に公開するデータを、特定領域に表示することを特徴とする請求項9に記載のデータ管理方法。

【請求項14】 前記特定領域にデータを移動することで、該データを前記指定工程により指定された端末に公開することを特徴とする請求項9に記載のデータ管理方法。

【手続補正3】

【補正対象書類名】明細書

【補正対象項目名】0004

【補正方法】変更

【補正内容】

【0004】

【発明が解決しようとする課題】しかしながら、上記従来例では、他のデータ処理装置に存在するデータへアクセスを試みる場合、利用者は当該データが存在するデータ処理装置を特定するなど、煩わしい操作が要求される。従って、データ処理装置ごとに固有の専門的な知識が必要とされ、かつ、計算機システムごとの複雑な操作をしなければならなかった。

【手続補正4】

【補正対象書類名】明細書

【補正対象項目名】0005

【補正方法】変更

【補正内容】

【0005】本発明は上記の問題に鑑みてなされたものであり、他の端末においてもアクセスを可能とするためのデータの公開操作を容易に行えるデータ管理方法及び装置を提供することにある。

【手続補正5】

【補正対象書類名】明細書

【補正対象項目名】0006

【補正方法】変更

【補正内容】

【0006】また、本発明の他の目的は、公開したデータについて所定の制約を課することを可能とし、公開したデータの管理を容易にすることにある。

【手続補正6】

【補正対象書類名】明細書

【補正対象項目名】0007

【補正方法】変更

【補正内容】

【0007】

【課題を解決するための手段】及び

【作用】上記の目的を達成するための本発明のデータ管理装置は以下の構成を備える。すなわち、アプリケーションプログラム等により作成されたデータをネットワーク上に接続された端末間で管理するデータ管理装置であって、前記データを記憶する記憶手段と、前記記憶手段に記憶されたデータから所望のデータを指示する指示手段と、前記指示手段により指示されたデータを複数の端末に公開するよう制御する制御手段とを備える。

【手続補正7】

【補正対象書類名】明細書

【補正対象項目名】0008

【補正方法】変更

【補正内容】

【0008】また、好ましくは、特定の端末を指定する指定手段を更に備え、前記制御手段は、前記指示手段により指示されたデータを前記指定手段により指定された特定の端末に公開するよう制御する。

【手続補正8】

【補正対象書類名】明細書

【補正対象項目名】0009

【補正方法】変更

【補正内容】

【0009】また、好ましくは、前記制御手段は、前記データを前記指定手段により指定された端末から参照或は更新を許可する。

【手続補正9】

【補正対象書類名】明細書

【補正対象項目名】0010

【補正方法】変更

【補正内容】

【0010】また、好ましくは、前記データに対する公開期限を設定する設定手段を更に備える。また、好ましくは、前記公開されたデータに対して、複写を制限する。

【手続補正10】

【補正対象書類名】明細書

【補正対象項目名】0011

【補正方法】変更

【補正内容】

【0011】また、好ましくは、前記指定手段により指定された端末に公開するデータを、特定領域に表示する。

【手続補正11】

【補正対象書類名】明細書

【補正対象項目名】0012

【補正方法】変更

【補正内容】

【0012】また、好ましくは、前記特定領域にデータを移動することで、該データを前記指定手段により指定された端末に公開する。

【手続補正 1 2】

【補正対象書類名】明細書

【補正対象項目名】0 2 4 7

【補正方法】変更

【補正内容】

【0 2 4 7】

【発明の効果】以上説明したように、本発明によれば、他の端末へのデータの公開操作を容易に行うことが可能

となる。また、本発明によれば、他の端末へデータを公開するにおいて、特定の端末に対してデータの公開を許可することが容易な操作で行える。また、本発明によれば、特定の端末に対して、データの公開期限を設定することができ、柔軟なデータ管理を行うことが可能となる。さらに、本発明によれば、公開データの複写を制限することができる。

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-313287

(43)Date of publication of application : 09.11.1999

-----  
-----  
(51)Int.Cl. H04N 5/92  
H04N 7/24  
// H03M 7/30

-----  
-----  
(21)Application number : 10-119428 (71)Applicant : BROTHER IND LTD

(22)Date of filing : 28.04.1998 (72)Inventor : TAGAWA NORIO  
TANIGUCHI TOSHINORI

-----  
-----  
(54) MOVING IMAGE DATA COMPRESSOR

(57)Abstract:

PROBLEM TO BE SOLVED: To provide the moving image data compressor that facilitates edit of non-real time data and widens the width of edit.

SOLUTION: Modules 311, 312,... corresponding to 'processing 1', 'processing 2',... are stored in a storage device 17. In the case that an icon 301 or 302 or the like corresponding to a desired processing is clicked or a name of processing is entered by a keyboard, the storage device 17 sends the module 311 or 312 or the like corresponding to the selected processing (rewinding or de-framing or the like) to a video signal processing section 15. The video signal processing section 15 stores the module sent from the storage device 17 and applies processing corresponding to the module to non-real time data received from a video file 7 and gives the resulting data to a compressor. Thus, desired processing is easily applied to the non-real time data